HUGHES INFORMATION TECHNOLOGY CORPORATION

ERRATA NOTICE

EOS Core System (ECS) Project Contract No. NAS5-60000

December 15, 1995

Document No.: 322-CD-001-002/414-CD-001-002

Title: Interim Release 1 Integration and Test Plan and Procedures for the ECS Project

Enclosed please find a change page for the subject documents. Please replace the page as follows:

Remove	Insert
iii and iv	iii and iv
1-1 and 1-2	1-1 and 1-2

If you have any questions, please contact our Data Management Office at (301) 925-0322.

322-CD-001-002/
414-CD-001-002

EOSDIS Core System Project

Interim Release 1 Integration and Test Plan and Procedures for the ECS Project

Final

December 1995

Hughes Information Technology Corporation
Upper Marlboro, Maryland

Interim Release 1 Integration and Test Plan and Procedures for the ECS Project

December 1995

Prepared Under Contract NAS5-60000
CDRL Item #054

SUBMITTED BY

C. M. Smith /s/	12/7/95
Chris Smith, Ir1 CCB Chairman EOSDIS Core System Project	Date

Hughes Information Technology Corporation
Upper Marlboro, Maryland

This page intentionally left blank.

Preface

This document is a final contract deliverable in accordance with the follow-up letter to 995-TR≠951-057, dated July 19, 1995 (Re: CDRL Items DID 319/DV1, DID 402/VE1, DID 322/DV3, and DID 414/VE1). It requires Government review and approval prior to acceptance and use. Changes to this document also require Government approval prior to acceptance and use. Changes to this document shall be made by document change notice (DCN) or by complete revision. Once approved, this document shall be under ECS Project Configuration Control.

Any questions should be addressed to:

Data Management Office
The ECS Project Office
Hughes Information Technology Corporation
1616 McCormick Drive
Upper Marlboro, Maryland 20774-5372

This page intentionally left blank.

Abstract

This document specifies the TRMM Infrastructure Interim Release 1 (Ir1) Test Plan and Procedures for the ECS Project. It includes descriptions of: the ECS test methodology, Build/Thread functional decomposition, detailed test cases, detailed test procedures, requirements traceability matrices between Level 3 Requirements-by-Release and Level 4 requirements to test cases, and descriptions of resources and test tools needed for these tests.

Keywords: integration, test, I&T, build, thread, interim, release, one, (Ir1), CSMS, SDPS, ECS, traceability, Level-4, Requirements-by-Release, (RBR), case(s), procedure(s), requirements, tools

This page intentionally left blank.

Change Information Page

List of Effective Pages	
Page Number	**Issue**
Title	Final
iii through xxvi	Final
1-1 and 1-2	Final
2-1 through 2-4	Final
3-1 through 3-14	Final
4-1 through 4-144	Final
5-1 through 5-368	Final
A-1 and A-2	Final
B-1 through B-4	Final
C-1 through C-90	Final
D-1 through D-12	Final
E-1 and E-2	Final
F-1 and F-2	Final
AB-1 through AB-4	Final

Document History			
Document Number	**Status/Issue**	**Publication Date**	**CCR Number**
322-CD-001-001/ 414-CD-001-001	Preliminary	August 1995	95-0614
322-CD-001-002/ 414-CD-001-002	Final	December 1995	95-0901

This page intentionally left blank.

Contents

Preface

Abstract

1. Introduction

2. Related Documentation

3. Ir1 Integration and Test Organization Overview

4. Ir1 Test Descriptions

5. DCE Infrastructure Thread (T1)

Figures

Tables

Appendix A. Test Tool Descriptions

Appendix B. Test Data Descriptions

Appendix C. Verification Traceability Matrix

Appendix D. Rationale Matrix for Unused PDR Test Cases

Appendix E. Test Case Formats, Notations and Naming Conventions

1. Introduction

1.1 Identification

This document is submitted as required by the Contract Data Requirements List (CDRL) item 054, DID 322/414 whose requirements are specified as a required deliverable under the Earth Observing System (EOS) Data and Information System (EOSDIS) Core System (ECS), Contract (NAS5-60000). This document is a final contract deliverable in accordance with the follow-up letter to 995-TR-951-057, dated July 19, 1995 (Re: CDRL Items DID 319/DV1, DID 402/VE1, DID 322/DV3, and DID 414/VE1).

1.2 Scope

This document defines the plan and procedures for integration, test, and verification of the TRMM Infrastructure Release, referred to as Ir1, for ECS. There is a separate document detailing the test activities for each proceeding release. The Ir1 Integration and Test Plan and Procedures applies to segment and system level verification activities. This document provides procedures to verify that the ECS complies with the Ir1 Level 3 Requirements-by-Release (RBRs) and Level 4 Functional Requirements. The roles and activities of the Ir1 Integration and Test Organization are described and high level schedules for performing these activities are addressed.

This document reflects the Technical Baseline submitted via contract correspondence no. ECS 194-00343.

1.3 Purpose

This Ir1 Integration and Test Plan and Procedures document describes the test, review, and analysis effort to be conducted by the Ir1 I&T organization. This document presents the overall processes and activities associated with verifying the release segment and system integration and test phases. The test plan provides an outline of the activities to be performed for Ir1 I&T, while the test procedures provide more detailed instructions for verification of the Ir1 release.

1.4 Status and Schedule

This document is a final contract deliverable in accordance with the follow-up letter to 995-TR-951-057, dated July 19, 1995 (Re: CDRL Items DID 319/DV1, DID 402/VE1, DID 322/DV3, and DID 414/VE1).

This submittal of DID 322/414 meets the milestone specified in the CDRL of NASA Contract NAS5-60000.

1.5 Organization

This document, which is based on Ir1 requirements, is organized into the following sections:

Section 1- Introduction, contains the identification, scope, purpose and objectives, status and schedule, and document organization.

Section 2- Related Documents, provides a bibliography of parent, applicable and reference documents for the Ir1 Integration and Test Document.

Section 3- Ir1 Integration and Test Overview, describes the process used to integrate and test the TRMM Infrastructure release.

Section 4- Ir1 Test Case Descriptions, describes the specific segment and system level thread and build tests, which will be used to verify the functionality of the release.

Section 5- Ir1 Test Procedures, details the steps required for executing the various thread and build tests.

Appendix A- Contains a list and brief description of the test tools needed for Ir1 Integration and Test.

Appendix B- Contains a list and brief description of the test data needed for Ir1 Integration and Test.

Appendix C- Contains the requirements traceability matrices, mapping test cases to Ir1 Level 3 RBRs and Level 4 requirements and the verification method for each requirement.

Appendix D- Contains traceability matrix, mapping PDR test cases that are no longer used to the new set of consolidated Ir1 tests.

Appendix E- Contains test case format descriptions, notations and naming conventions as used in the submission of this document.

Appendix F Contains templates for the test logs and procedures.

Abbreviations and Acronyms Contains a listing of abbreviations and acronyms used in this Document.

2. Related Documentation

2.1 Parent Documents

The parent documents are the documents from which this Ir1 Integration and Test Plan scope and content are derived.

101-CD-001-004-	Project Management Plan for the EOSDIS Core System, Revision 1, DCN No. 01
194-107-MG1-XXX	Level 1 Master Schedule for the ECS Project
194-201-SE1-001	Systems Engineering Plan for the ECS Project
301-CD-002-003	System Implementation Plan for the ECS Project
304-CD-002-002-	Science and Data Processing Segment (SDPS) Requirements Specification for the ECS Project, Final
304-CD-003-002-	Communications and System Management Segment (CSMS) Requirements Specification for the ECS Project, Final
319-CD-002-002-	SDPS Integration and Test Plan for the ECS Project, Volume 1: Interim Release 1 (Ir-1), Final
319-CD-003-003-	CSMS Integration and Test Plan for the ECS Project, Volume 1: Interim Release 1 (Ir-1), Final
402-CD-001-002-	System Integration and Test Plan for the ECS Project, Volume 1: Interim Release 1 (Ir-1), Final
194-501-PA1-001	Performance Assurance Implementation Plan for the ECS Project
222-WP-001-002	Mission Statement for Interim Release One for the ECS Project
423-41-01-	Goddard Space Flight Center, EOSDIS Core System (ECS) Statement of Work
423-41-02-	Goddard Space Flight Center, Functional and Performance Requirements Specification (F&PRS) for the Earth Observing System Data and Information System (EOSDIS) Core System (ECS)
423-41-03-	Goddard Space Flight Center, EOSDIS Core System (ECS) Contract Data Requirements Document

2.2 Applicable Documents

The following documents are referenced within this Ir1 Integration and Test Plan, or are directly applicable, or contain policies or other directive matters that are binding upon the content of this volume.

194-207-SE1-001	System Design Specification for the ECS Project
194-401-VE1-002	Verification Plan for the ECS Project, Final

2.3 Information Documents

2.3.1 Information Documents Referenced

The following documents are referenced herein and amplify or clarify the information presented in this document. These documents are not binding on the content of the ECS Ir1 Integration and Test Plan.

102-CD-001-004	Development Configuration Management Plan for the ECS Project
193-103-MG3-001	Configuration Management Procedures for the ECS Project
108-CD-000-015	Intermediate Logic Network (ILN) Diagram for the ECS Project
305-CD-002-002-	Science Data Processing Segment (SDPS) Design Specification for the ECS Project
305-CD-003-002-	Communications and System Management (CSMS) Design Specification for the ECS Project
308-CD-001-005	Software Development Plan for the ECS project

2.3.2 Information Documents Not Referenced

The following documents, although not referenced herein and/or not directly applicable, do amplify or clarify the information presented in this document. These documents are not binding on the content of the ECS Ir1 Integration and Test Plan.

104-CD-001-004	Data Management Plan for the ECS Project
193-105-MG3-001	Data Management Procedures for the ECS Project
209-CD-007-003-	Interface Control Document Between EOSDIS Core System (ECS) and TRMM Science Data and Information System (TSDIS)
209-CD-008-002-	Interface Control Document Between EOSDIS Core System (ECS) and the Goddard Space Flight Center (GSFC) Distributed Active Archive Center (DAAC) for the ECS Project

209-CD-009-002 Interface Control Document Between EOSDIS Core System (ECS) and the Marshall Space Flight Center (MSFC) Distributed Active Archive Center (DAAC) for the ECS Project

404-CD-001-001 Procedure for Control of Unscheduled Activities During Verification for the ECS Project

505-41-14 - Goddard Space Flight Center, (TRMM-490-152) Interface Requirements Document Between EOSDIS Core System (ECS) and Tropical Rainfall Measuring Mission (TRMM) Ground System

560-203.103- Interface Control Document Between the Sensor Data Processing Facility (SDPF) and the Tropical Rainfall Measuring Mission (TRMM) Consumers

This page intentionally left blank.

3. Ir1 Integration and Test Organization Overview

This section contains an overview of the approach taken by the Ir1 Integration and Test organization to ensure complete and thorough testing at the segment and system levels. Included is information concerning the Ir1 I&T environment, schedules and verification activities and responsibilities.

3.1 Ir1 I&T Organization and the ECS Environment

3.1.1 Ir1 Functional Overview

The Ir1 system has four primary functional areas. Further detail can be found in the Mission Statement for Interim Release One for the ECS Project (Document 222-WP-001-002). Appendix B herein includes the Ir1 ECS DCE Cell Topology diagram and the Ir1 External Interface Diagram (both obtained from Document 222-WP-001-002).

Science Software Integration and Test (SSI&T). Ir1 provides an environment for the early integration and testing of science software from the EOS AM-1 instrument teams and TRMM's CERES and LIS instrument teams. Early SSI&T gives the instrument teams, DAAC personnel and the ECS development team the ability to test the portability of the science software using test data provided by the instrument teams. The following tools are provided to support the SSI&T capabilities:

a. SCF and DAAC versions of SDP Toolkit

b. Compilers: C, C++, FORTRAN 77 and 90, ADA (LaRC only)

c. File comparison utility

d. Static and dynamic code checkers for standards compliance

e. Profiling Tools for resource monitoring

f. Product visualization/graphics Tool

g. Document viewing tools

h. Math, graphics, and statistics libraries

i. Software configuration management tool

Early TRMM Interface Testing. Ir1 provides capabilities for early functional testing of TRMM-ECS interfaces among several facilities including the Sensor Data Processing Facility (SDPF), the TRMM Science Data and Information System (TSDIS), NOAA/NESDIS, the Data Assimilation Office (DAO), and three DAACs - LaRC, GSFC and MSFC. The following Ir1 interfaces will be delivered:

a. TRMM Data Ingest Interface

Ir1 provides capabilities for supporting TRMM ingest interface testing. The system supports the testing of the automated network ingest interface with SPDF and TSDIS. This testing includes the exchange of security authentication messages, the verification of message protocols, and the verification of the ingest file transfer capability. Ir1 also supports the testing of the polling ancillary data ingest interface with NESDIS and DAO.

b. TRMM Data Retrieval Interface

Ir1 provides the capability to support the testing of the Data Server interface with TSDIS. This testing includes the exchange of security authentication messages, the verification of message protocols, and the verification of the capability to transfer files from the Data Server subsystem to TSDIS.

c. External Interface Gateway

The Ingest and Data Server interfaces are designed to communicate with external clients using communication services provided by the OSF Distributed Computing Environment (DCE). TRMM and SDPF use protocols based on UNIX sockets, not DCE. Ir1 provides a communications gateway which allows these external clients to interface with Ir1 using UNIX socket calls.

Planning and Processing. Ir1 provides basic capabilities for managing science data processing tasks. It supports manual capabilities for generating processing plans and for process initiation and control. It supports process execution profiling and diagnostic reports on Sun and SGI processors (note: the Sun is a 32-bit processor, while the SGI can operate in either a 32-bit or 64-bit mode). Ir1 also provides a prototype scheduling capability that is based on a COTS scheduler. The scheduler will enable the execution of multiple PGEs in sequence, using pre-staged test data.

ECS Infrastructure. Ir1 provides an early implementation of communication and system management services and verifies the communications and management infrastructure. The infrastructure consists of the following:

a. Basic naming and directory, time, thread and security services (DCE-based)

b. File transfer capability, email, bulletin board, event logger, virtual terminal (telnet and X)

c. EDF-based framework for system management and DAAC performance monitoring

d.- Site-based COTS, SNMP agents for hosts, and network components as provided by Version 0

e.- Authentication (DCE account management), authorization (host-level account management), router-based security (address table management) as provided by Version 0 networks

f. Site-based office automation tools

g.- Configuration management tools for science software (at the DAACs) and development software at the EDF

3.1.2 Ir1 I&T Organization and Relationship to other Test Groups

The Ir1 I&T organization is responsible for integration and test at the segment and system levels for the Ir1 Release. This includes: acceptance of software components upon completion of unit level testing, integration of these components into segment subsystems, complete and thorough testing of the integrated system, and recording and reporting of any problems encountered during testing. The integrated release is tested against Level 3 Requirements-by-Release (RBRs) and Level 4 requirements as baselined in the Ir1 CCB. The Ir1 I&T organization is responsible for verifying functional components, subsystem interfaces, and external interfaces to support the delivery of the Ir1 release.

The Ir1 I&T organization interacts with and supports other ECS and independent test organizations. This includes the Quality Office, SMO (both ECS and ESDIS), and EOSDIS Independent Verification and Validation (IV&V) Contractor. The IV&V contractor monitors ECS verification activities, and conducts independent tests to verify compliance with the Level 3 requirements documented in the Functional and Performance Requirements Specification (Goddard document number 423-41-02). Figure 3.1-1 illustrates the interactions between the Ir1 I&T team and other organizations.

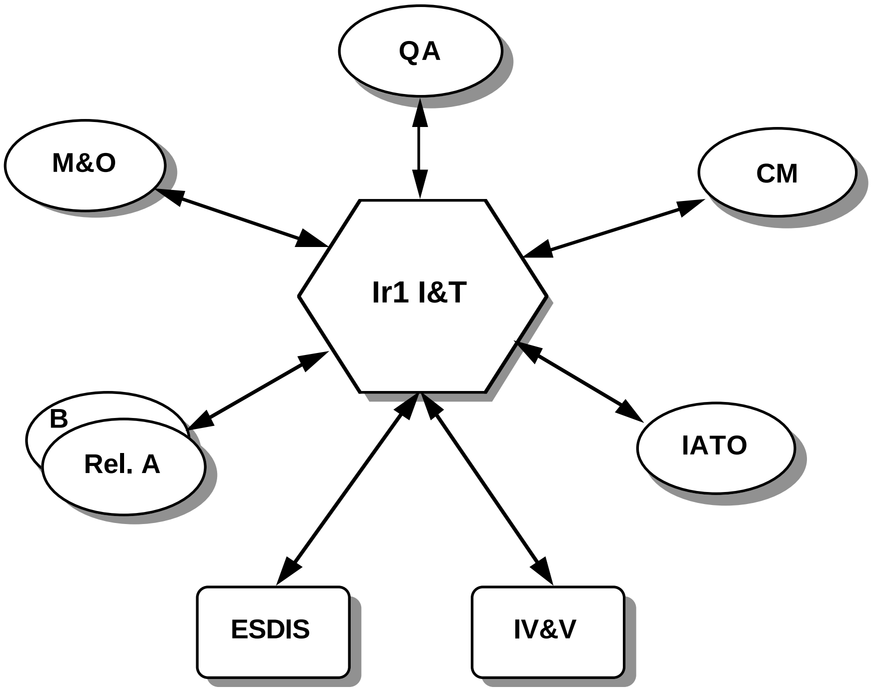

Figure 3.1-1. Ir1 I&T Interactions

The Quality Office assists in identifying training needs of test personnel and schedules formal training. The Quality Office conducts requirements traceability audits during the Implementation and Integration and Test phases as each test case is completed and evaluated. They are responsible for monitoring the hardware inspection and unit-level verification procedures and verify release test plans for completeness. They also validate release integration and tests and test results. The Quality Office participates in release test implementation, reviews, and analysis. They are also responsible for monitoring the life cycle of the nonconformance reports and participating in the final decision on product acceptability.

Upon completion of Ir1 testing, a Consent to Ship Review (CSR) will be held. Since Ir1 is developed on the incremental path, Independent Acceptance Test activities (through the IATO) will not occur. Instead, the Ir1 I&T team will conduct the release deployment, installation, and on-site testing. Further information on these activities is described in Section 3.6 (Ir1 I&T Schedule). Once on-site tests are completed, the Independent Verification and Validation (IV&V) contractor will provide independent assessments of the functionality and performance of the ECS Ir1 release. The IV&V contractor is responsible for the validation of the ECS Level 3 requirements. They are also responsible for the reporting and tracking of nonconformances identified during this phase of testing.

Figure 3.1-2 summarizes the overall test activities for Ir1 under the new ECS release organizational structure. The EDF at Landover will be the central location for all Ir1 integration and test activities (both segment and system). Certain external interfaces will be simulated using SDPF provided tools modified for ECS use. Developers and testers will utilize the EDF DAAC to port, integrate and test components and subsystems. As mentioned earlier, the Ir1 I&T team will also be responsible for system deployment and on-site testing in lieu of formal IATO activities. Interface testing at these sites will primarily rely on the simulators used at the EDF in conjunction with available resources from TSDIS and SDPF that can be scheduled during the short period of ECS Ir1 site testing.

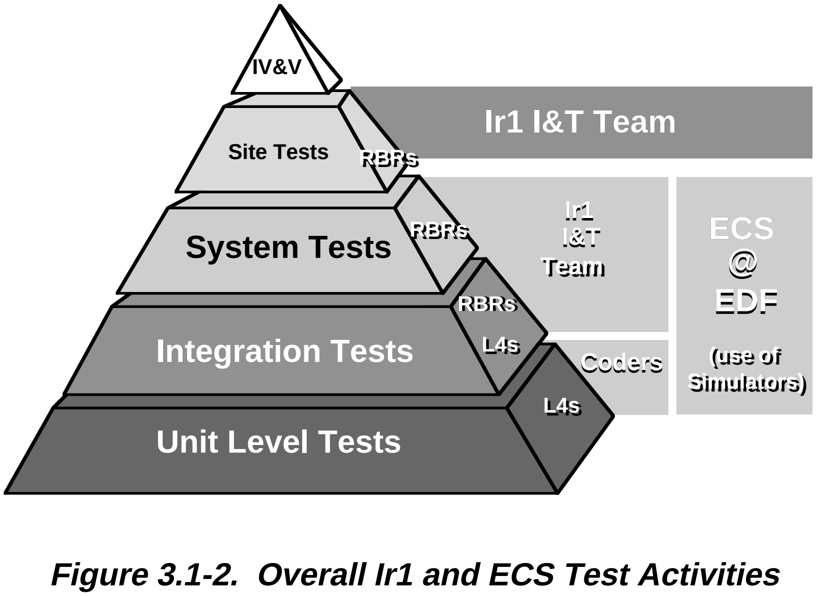

Figure 3.1-2. Overall Ir1 and ECS Test Activities

Additional ECS test methodologies and organizational aspects reflecting the new ECS release organization will be documented in the updated Verification Plan for the ECS Project (194-401-VE1-002).

3.2 Ir1 I&T Test Approach

The Ir1 I&T organization will integrate and verify release functionality on an incremental basis. As incremental integration and testing proceeds, larger portions of the system are assembled thus being available for final testing. As unit level testing on components is completed by the Ir1 developers, the Ir1 I&T organization will incrementally assemble lower-level functionality into progressively higher levels until ultimately subsystems are completely integrated into the release/system. Functional components that are integrated are threads, and the result of combining threads (and/or builds) is a build. The Ir1 functional tests are requirement driven, thus verifying the Level 3 RBRs and Level 4 requirements allocated to Ir1.

A number of project instructions (PIs) facilitate the ongoing integration and test efforts for Ir1. These PIs are listed below:

CM-1-025	ECS Project Instruction: Software Development Handbook
CM-1-023	ECS Project Instruction: Software Build Process
SD-1-013	ECS Project Instruction: COTS Process Model
SD-1-014	ECS Project Instruction: Nonconformance Reporting
SE-1-002-	ECS Project Instruction: ECS Development Facility (EDF) Configuration Control

3.2.1 Ir1 Build/Thread Methodology

The build/thread concept, which is based on the incremental aggregation of functions, is used to plan Ir1 I&T activities. An Ir1 thread is the set of components (software CIs, hardware and data) and operational procedures that implement a function or set of related functions at that level. Threads are tested individually to facilitate the verification of requirements allocated to them and to isolate software problems. A build is an assemblage of threads (and/or builds) that produces a gradual buildup of system capabilities. This orderly progression of combining lower level software and/or hardware items to form higher level items with broader capability is the basis for the Ir1 integration activities. Tests at the build level combine regression testing of capabilities that made up the build, along with additional system flavor tests that address the broader span of capabilities as they are integrated at the build level.

The Build/Thread diagram for Ir1 is presented in Figure 3.2-1. Threads and builds are defined by examining the Ir1 capabilities and requirements. In addition, schedule dependencies relating to code hand-offs from the developers to the Ir1 I&T team, impact the chronological order in which the integration process will proceed. The build/thread diagram for Ir1 acts as a framework for development of test cases. These test cases provide the basis for development of step-by-step test procedures, as described herein.

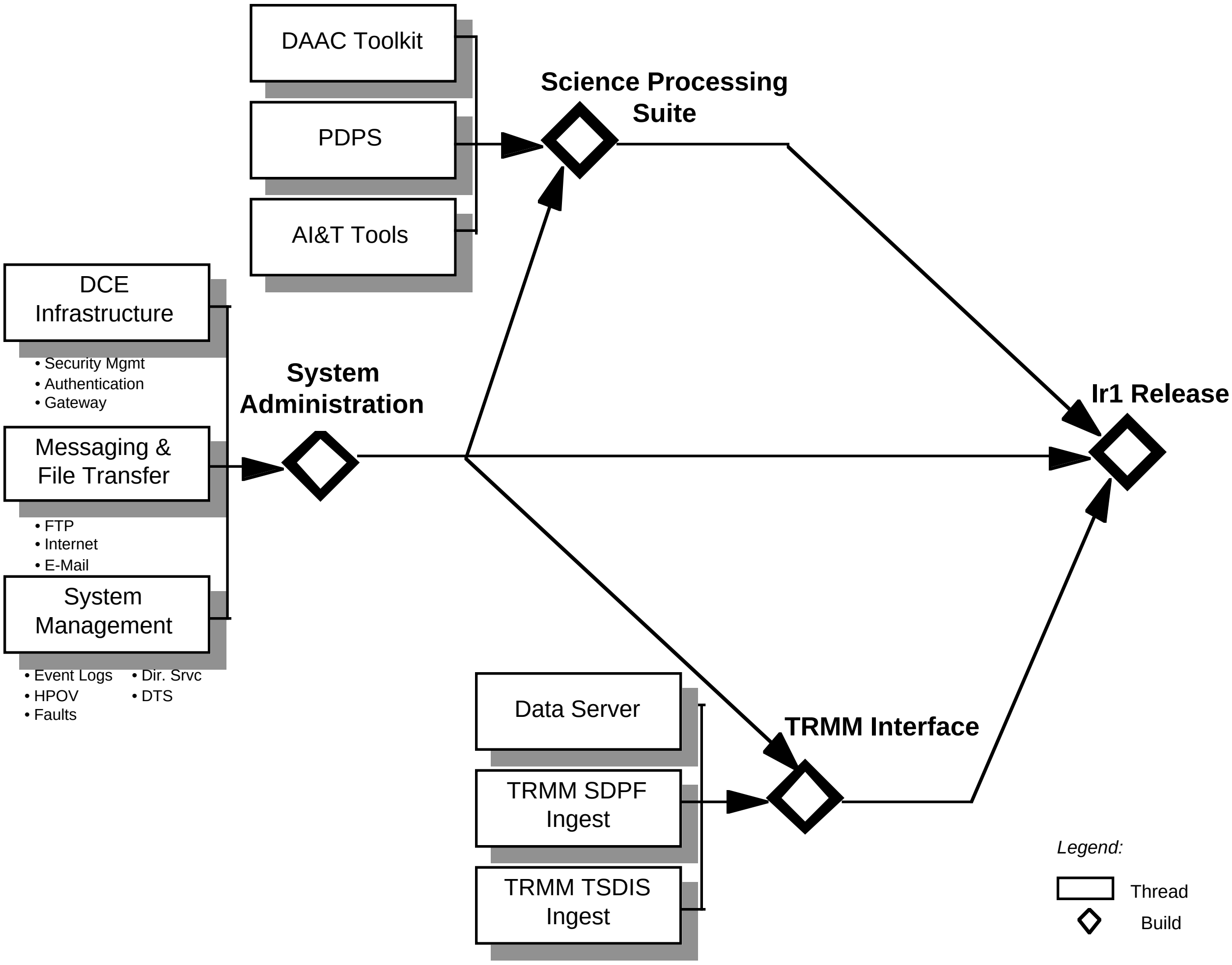

Figure 3.2-1. Interim Release One Build/Thread Diagram

3.3 Ir1 I&T Verification Process

The following sections define responsibilities and activities of the Ir1 I&T organization. The following topics are addressed: definition of verification methods, post test analysis, regression testing, and verification resources.

3.3.1 Verification Methods

The four verification methods used for Ir1 I&T activities include: inspection, analysis, demonstration, and test.

Definitions for these methods, as defined in the ECS Verification Plan (Contract # 194-401-VE1 002) are as follows:

a.- Inspection - The visual, manual examination of the verification item and comparison to the applicable requirement or other compliance documentation, such as engineering drawings.

b.- Analysis - Technical or mathematical evaluation based on calculation, interpolation, or other analytical methods. Analysis involves the processing of accumulated data obtained from other verification methods.

c.- Demonstration - Observation of the functional operation of the verification item in a controlled environment to yield qualitative results without the use of elaborate instrumentation, procedure, or special test equipment.

d.- Test - A procedure or action taken to determine under real or simulated conditions the capabilities, limitations, characteristics, effectiveness, reliability, or suitability of a material, device, system, or method.

Each Ir1 allocated requirement will be tested and verified by one or more of the above methods. A requirements traceability matrix, mapping test cases to requirements, will include the method(s) of verification. These matrices are provided in Appendix C of this document.

3.3.2 Post Test Analysis

Post-test analysis includes data reduction and comparison of actual results against expected results. Any post test analysis required for Ir1 I&T activities will be performed by the Ir1 I&T organization with support from the user and development communities when appropriate. Methods for performing post-test analysis will be documented in the Ir1 Integration and Test Procedures on a test by test basis. Post-test analysis results will be documented in Ir1 I&T Report (CDRL 324/405). Data, data logs, event logs and any other test output required for post test analysis will be captured and stored under CM control.

3.3.3 Regression Testing

Regression testing is supplemental testing performed at any time upon any build or thread during Ir1 I&T testing to ensure that existing software is not adversely affected by modified or new software. The Ir1 I&T organization is responsible for planning, documenting, executing and reporting all regression testing. Automated test tools are used, when practical, for regression testing by the Ir1 I&T organization. This ensures that regression tests duplicate initial test procedures.

Ir1 Regression Testing will occur as a result of:

- software changes
- hardware changes
- operational enhancements
- integration of two or more builds
- new versions delivered after unit level testing

The Ir1 I&T organization is responsible for reporting any discrepancies encountered during regression testing. Discrepancies resulting from future ESDIS and/or IV&V testing which result in software changes, will be regression tested at the EDF (or site in certain cases) by the Ir1 sustaining engineering organization.

3.3.4 Verification Resources

This section introduces and identifies the required resources necessary to accomplish the various Ir1 I&T activities. Included are identification of test location, hardware and software configurations. Also discussed are the use of automated test tools, discrepancy reporting, and the role of CM in Ir1 I&T activities.

3.3.4.1 Testing Facilities

The EDF, located at the ECS facility in Landover, Md., has been designated as the testing facility for Ir1 I&T activities. This facility will be shared with the Ir1 developers. ECS will be solely responsible for the test environment. This includes installation, initial checkout and startup, upgrades/version control, access control, and maintenance. In addition, the Ir1 I&T organization will conduct on-site tests as part of the deployment process. These tests will utilize the installed facilities at the DAACs designated for Ir1.

3.3.4.1.1 Hardware Items

The hardware CIs available for the Ir1 time frame will be configured as the EDF DAAC for this release. The hardware will also be used to emulate all communications interfaces available for Ir1.

3.3.4.1.2 Software Items

All COTS software packages will be installed and configured on their respective hosts/servers. All custom developed software CSCIs will be retrieved from CM using the process outlined in the respective ECS Project Instructions (see Section 2.3.1).

3.3.4.2 Test Tools and Test Data

The Ir1 I&T organization uses test tools for test development, test execution, and test management. Whenever possible, test tools from the unit development and unit test environments will be reused. Additional test tools consist of COTS products or are developed by the Ir1 I&T organization. For a complete listing and description of the test tools please see Appendix A. All test cases which require an external interface to support the Ingest and Data Server will utilize the simulators described in Appendix A. In all cases where testing of the DAACs is mentioned, we are referring to the ECS deployed facilities.

During Ir1 test development, test tools will be used to develop test scripts and map requirements to test cases. The Ir1 I&T organization will use the ECS selected tools for test script development and requirements traceability. The ECS selected capture/playback tools will be used to aid in the development of test procedures. The ECS selected Requirements & Traceability Management (RTM) tool to be used for mapping Ir1 I&T test cases to requirements. This tool is used for all releases.   A unique number is assigned to each build/thread and test case. This number is then used to identify the build/thread test case in RTM.

During Ir1 test execution, test tools will be used to simulate data and interfaces, decipher and monitor the data transmitted over the network and facilitate the execution of test procedures. Data interface simulators and user emulators are needed for interfaces that do not yet exist or are not yet mature enough for test use. Test data generators to simulate various data transmissions may be required. Additional tools for test execution include: capture playback tools, drivers, interface simulators, user emulators and data generators. Network Analyzers are used to monitor and analyze the data that is transmitted over the network. Capture Playback Tools are used for replaying user sessions for regression testing, and to emulate multiple virtual sessions for system load and performance tests.

Test management tools record test results and aid in test result data analysis. These tools include loggers and other recording devices and reduction and analysis programs. File comparison utilities may be needed to compare data output with data input. The event logs gathered by management subsystem tools and agents will be used to aid in the data analysis phase of testing.

The capture playback tool selected was XRunner, and the user emulation tool selected was LoadRunner. Both of these tools were developed by Mercury Interactive Corporation. The selection is defined in the EDS/ECS Source Evaluation Recommendation for Automated Test Tool Procurement (RFP #013). Hewlett Packard OpenView is the selected Enterprise Management Framework that will be used to monitor the network activity, loads, etc. during the testing phase. The selection is documented in the EDS/ECS Source Evaluation Recommendation for Enterprise Management Framework (July 22, 1994).

Since there is no operational data available for Ir1 testing, test data is either provided from organizations holding appropriate data (e.g., TSDIS) or must be provided by a data generator. As ECS matures in future releases, the types and formats needed to satisfy test case needs will differ. Test data needed for Ir1 is described in Appendix B of this document.

3.3.4.3 Discrepancy Reporting and Resolution

Ir1 I&T is required to report any nonconformance with Level 3 RBRs and Level 4 requirements encountered during the I&T activities. The Ir1 I&T organization will use the ECS selected COTS tool for tracking nonconformances (DDTS). It is the responsibility of the Ir1 I&T organization to assure that all testers are trained to use the Nonconformance Reporting and Corrective Action (NRCA) system. The Ir1 I&T staff will have the proper authority and access to the NRCA tool before any Ir1 I&T activities begin. It is the responsibility of each tester to properly enter all discrepancies encountered during testing into the NRCA system. Once the discrepancy is corrected, regression testing is done to make sure no new problems have been introduced by the fix. If necessary, the tester will develop additional tests to ensure the problem is satisfactorily corrected. Quality Assurance representatives are responsible for audits to ensure reported nonconformances are resolved and properly verified. The following table describes the severity levels used in the NRCA system (additional detail can be found in the Nonconformance Reporting PI - see Section 2.3.1 for appropriate reference).

Table 3.3-1. NCR Severity Level Definitions

Severity Level	Definition
1	Catastrophic deficiency without work around that causes total failure or unrecoverable data loss.
2	Deficiency which severely impairs functionality. Work around might exist but is unsatisfactory.
3	Deficiency that causes failure of noncritical system aspects. There is a reasonably satisfactory work around.
4	Deficiency of minor significance. Work around exists or, if not, the impairment is slight.
5	Very minor defect. Work around exists or the problem can be ignored.

3.3.4.4 Test Items Under Configuration Control

ECS Ir1 I&T organization test documents, software and hardware configurations under test, test data sets, and software and hardware tools used for testing are maintained by CM. Ir1 I&T will use the ECS selected COTS tool for configuration management control. It is the responsibility of the Ir1 I&T organization to train all testers to use the CM tool. The Ir1 I&T staff will have the proper authority and access to unit tested components using the CM tool before any Ir1 I&T activities begin. Unit-tested components entered in the CM system are accessed by the Ir1 testers. These components are verified and integrated by the Ir1 I&T staff. Upon successful completion of Ir1 I&T verification activities, the CM organization will create a delivery version for the release.

Fixes to the discrepancies found during Ir1 I&T activities (see section 3.3.4.3) are tracked in the CM tool through the product changes and versions that result from correcting discrepancies. Code modified on so called "bug-fix" branches will then be merged into the main branch prior to rebuilding the binaries. See appropriate ECS Project Instruction in Section 2.3.1.

3.4 Ir1 I&T Roles and Responsibilities

The Ir1 I&T test team roles include the following test positions and their corresponding responsibilities. Figure 3.4-1 depicts the Ir1 I&T team organization.

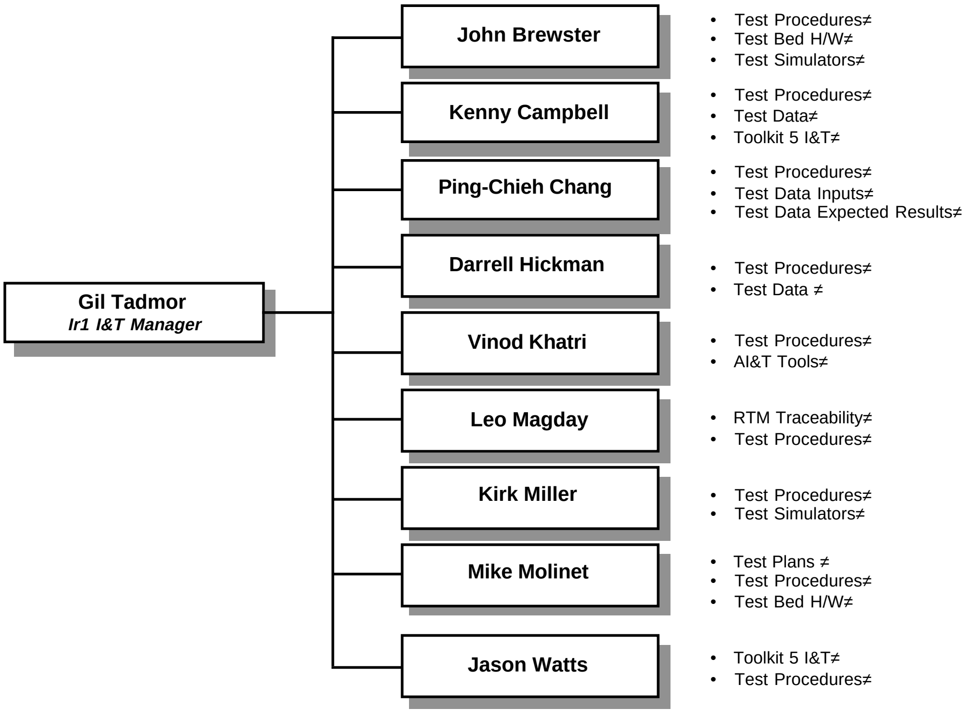

Figure 3.4-1. Ir1 I&T Organization

Test Conductor - An Ir1 I&T member to conduct test execution. This person is responsible for establishing a stable and well-defined test configuration before testing takes place. This person is also responsible for collecting test outputs and recording test results. Any problems encountered during testing are entered into the NRCA System by the test conductor.

Test Participants - Ir1 I&T members and members of the development organization to perform subsystem integration and support test execution. Other supporting organizations include Maintenance and Operations (M&O) and Configuration and Data Management (CM). The ECS maintenance and operations organization will support the test members in the installation and configuration of the test environment and will support the test team if any system faults are encountered during testing. This would include such instances as computer software or hardware failures which cause the test configuration to be corrupted. M&O will be responsible for reconfiguring the system as needed to continue testing. CM will provide a controlled environment for the storing and maintaining of information about the test environment including hardware, software and test tool environments. CM also stores and catalogs test documents and test input data and output data.

Test Witnesses - Individuals invited to directly observe test conduct. This will include members from the ECS IATO and ESDIS IV&V.

Test Monitors - The Quality Assurance organization is responsible for reviewing test data, materials, and documentation. These individuals need not be present during test conduct.

Test Manager - Individual responsible for managing the Ir1 I&T team, coordinating activities and schedules, and assuring successful completion of all verification tasks prior to delivery.

3.5 Ir1 I&T Release Testing

Ir1 I&T verification activities occur during the entire progression of the release. Initial preparatory tasks include requirements traceability, test case descriptions, test procedure development, test bed configuration, test execution, and test reporting. These activities begin at the earliest stages of the release as possible using the model depicted in Figure 3.5-1. Due to Ir1's development on the incremental track, the only formal review scheduled is the Consent to Ship Review (CSR) to be held following the completion of Ir1 I&T. Figure 3.5-2 provides further insight as to the specific activities occurring during the I&T phase. See Section 3.6 (Ir1 I&T Schedule) for further detail on Ir1 I&T activities leading to and following CSR.

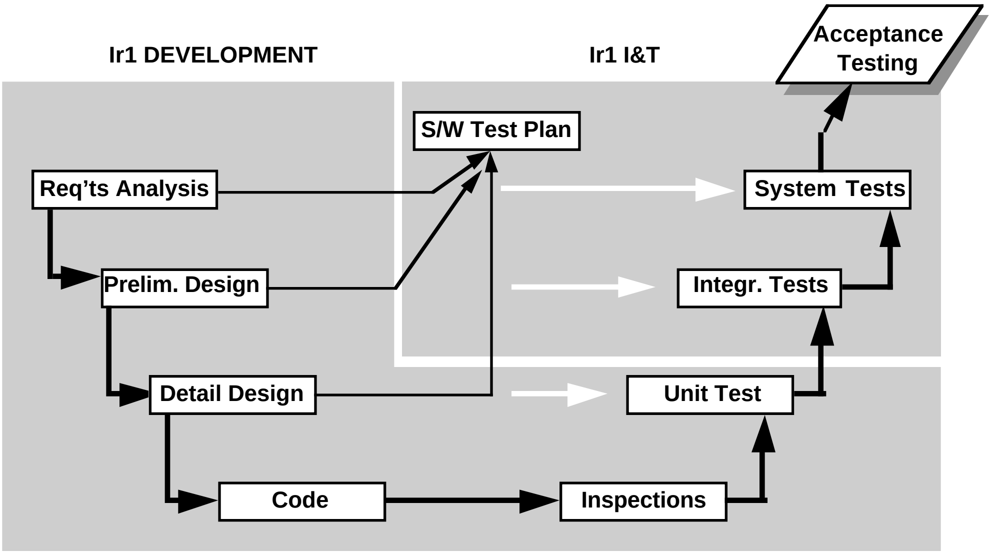

Figure 3.5-1. V-Shaped Lifecycle Model

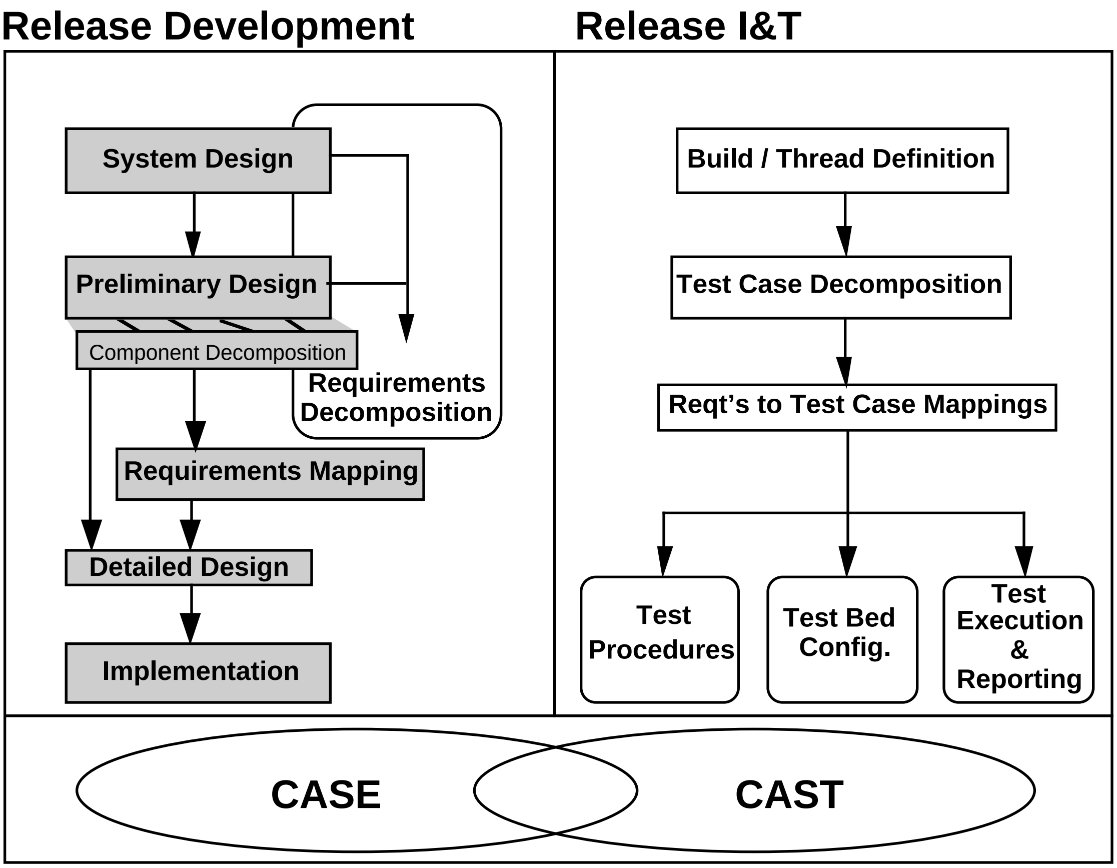

Figure 3.5-2. Ir1 I&T Activities

3.6 Ir1 I&T Schedule Overview

The Figure 3.6-1 illustrates the Ir1 I&T organizational activities during the release through deployment and on-site testing at the DAACs. Specific dates for these activities and milestones can be found in the ECS Ir1 Intermediate Logic Network (ILN) (CDRL 108-CD-000-015).

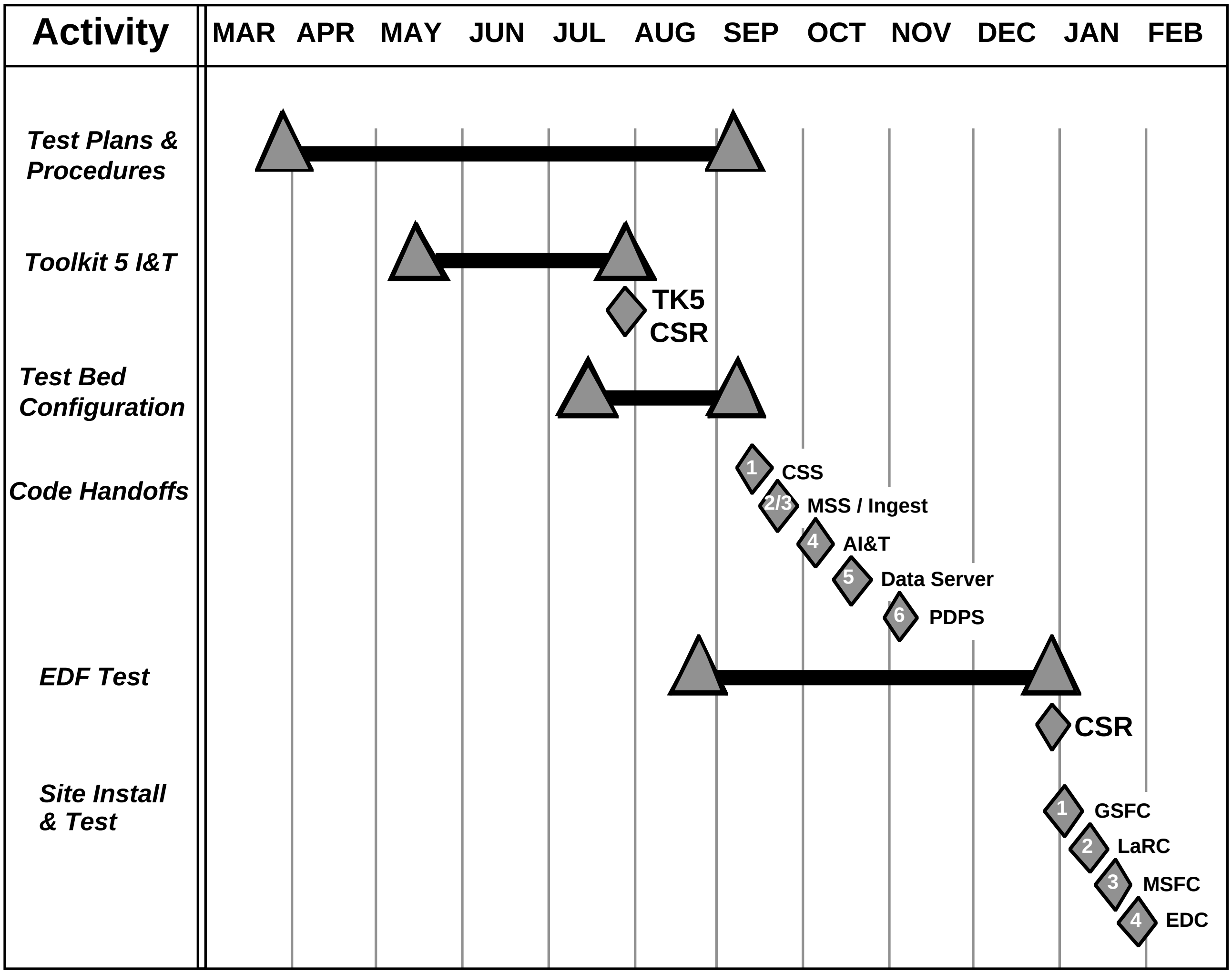

Figure 3.6-1. High Level I&T Schedule for Ir-1

4. Ir1 Test Descriptions

The following sections identify all Ir1 I&T test cases. First, threads are identified. Threads are the aggregation of unit tested components (HWCIs, CSCIs, CSUs, COTS software). Each thread demonstrates an Ir1 function. Builds are the integration of threads and are identified after each series of threads and/or builds which make up the build. Test cases are identified for each thread and build. The primary objective of each test case is to verify and evaluate the capabilities of each function as stated in Level 3 RBRs and 4 requirements.

4.1 Ir1 Infrastructure Thread Test

This thread demonstrates the functionality of providing Unix and DCE authentication service for IR1. This service includes secure DCE logon/logoff and maintenance of the user authentication directory. This service is provided via DCE security services, as configured in the single Ir1 cell.

The objective of this thread is to prove that user authentication is valid, reliable, and secure and that the user authentication directories can be maintained reliably. In addition, the Ir1 Gateway will be exercised, and systems inspections will be performed.

Special resources required for this thread test include:

- Network analyzer
- XRunner (optional)

4.1.1 Test Case 1.1: Systems Inspections (TC017.001)

The objective of the Systems Inspections test is to verify the various Level 4 systems requirements allocated to Ir1 through inspection of one or more of the following:

- Vendor Documentation
- Product Specifications
- ECS Procurement Documentation
- Inspection of physical hardware components in the EDF during their configuration

<u>Test Inputs</u>

Inputs to this test case include all applicable documentation required for the inspections (as listed above).

<u>Test Steps</u>

Inspect the available documentation and/or the actual hardware component in the EDF.

Test Outputs

The expected outputs include hardware specifications and procurement documentation that describes the component configuration.

Success Criteria

This test will be deemed successful when Ir1 sytems components are in compliance with the respective requirements through inspection of the above sources.

Assumptions and Constraints

None.

4.1.2 Test Case 1.2: User Authentication (TC003.001)

This test case demonstrates a successful DCE logon capability and the ability to interface with the authentication database.

Test Inputs

A valid DCE ID and a valid password.

Test Steps

Perform a DCE log on and log off using three valid ID/Password combinations.

Repeat for guest account.

Gather the information using a network analyzer, verify that the DCE password is not readable over the network.

Test Outputs

Screen outputs showing the success or failure of the DCE logon/logoff attempts. Network monitor output showing the data transmitted between the client and server.

Success Criteria

Three successful logon and logoff attempts with each event occurring in under 15 seconds. No in the clear password data on the network.

Assumptions and Constraints:

None.

4.1.3 Test Case 1.3: Failed User Authentication (TC003.002)

This test case demonstrates the ability to detect an invalid logon (UNIX and DCE).

Test Inputs

ID	Password
Valid ID	Invalid Password
Invalid ID	Valid Password
Invalid ID	Invalid Password
Null ID	No Password Prompt
Valid ID	Null Password

Test Steps

Perform log ons using the above combinations.

Attempt each logon 3 times. (For test purposes logon termination will be set at 3 failed attempts.)

Using a network analyzer, verify that the DCE password was not transmitted in clear text.

Test Outputs

Screen outputs showing the success or failure of the logon/logoff attempts. General response times of each logon and logoff event. Network monitor output showing the data transmitted between the client and server. Event log data.

Success Criteria

All attempts at logon rejected and event log data that shows each failed logon with the appropriate data such as IDs attempted. Logon process terminated after three unsuccessful attempts. No error messages given that would aid in determining a valid ID password combination. The Data Encryption Standard (DES) for encryption and decryption of data is supported.

Assumptions and Constraints:

Logon termination set at three unsuccessful attempts. If an invalid logon occurs the error message provided will not indicate which input (ID or password) was invalid. For DCE logons when an invalid (unregistered) user id is attempted, you will not be prompted for a password.

4.1.4 Test Case 1.4: User Password Change (TC003.003)

This test case demonstrates the ability of a user to change their DCE password.

Test Inputs

A valid DCE ID and password.

A valid DCE password, new password, and repeated new password.

Test Steps

DCE log on.

Change DCE password.

DCE log off.

Dce log on with the new password.

DCE log off.

Attempt DCE log on with old password.

Test Outputs

Screen outputs showing the success or failure, when applicable, of the DCE logon/logoff attempts.

Success Criteria

Successful DCE logon with the changed password. Old password logon fails.

Assumptions and Constraints:

If it is necessary for XRunner to enter the existing and new passwords, the passwords will be visible in the XRunner script. To minimize the risk of intrusion, the password will be stored as a variable in the script, and will be assigned a value just prior to the test. Test case 1.2 must pass prior to attempting this test case.

4.1.5 Test Case 1.5: User Password Reset (TC003.004)

This test case demonstrates the ability of a DCE administrator to reset a user password.

Test Inputs

A DCE password reset for a specific ID.

The valid DCE ID and reset password.

A reset DCE password, new password, and repeated new password.

The valid DCE ID and new password.

Test Steps

Reset a DCE user password using an administration ID.

Perform a DCE user logon with the ID and reset password.

Perform a DCE password change.

Log off.

Log on with the new password.

Log off.

Attempt logon with reset password.

Attempt logon with old password.

Test Outputs

Screen outputs showing the success or failure of the DCE logon/logoff attempts.

Success Criteria

Successful DCE logon with the reset password.

A forced password change.

Successful DCE logon with the changed password.

Failed logon with reset and old password.

Assumptions and Constraints:

Test case 1.2 must pass prior to attempting this test case.

4.1.6 Test Case 1.6: Security Registry Maintenance (TC003.005)

This test case demonstrates the ability of a DCE administrator to maintain the security registry.

Test Inputs

A valid DCE administrator ID and password.

Valid add, change and delete registry commands.

Valid access control privileges.

Test Steps

Execute a DCE security administrator logon.

Perform add, change and delete commands to the security registry.

Verify that the user accounts contain user name, password, group and user identification code, login directory and command line interpreter.

Log off.

Verify the history log file.

Perform DCE logon with user ID.

Perform add, change and delete commands to the security registry.

Verify that the Security Management Application Service provides the capability to set, maintain, and update the access control (i.e., read, write, execute privileges) information for ECS resources.

Log off.

Execute a logon to a server.

Perform change of DCE password for the server.

Verify that the server recognizes the new password, and ensure that there is a valid DCE login prompt.

Log off.

Verify that the Security Management Application Service provides the capability to set, maintain, and update the access control (i.e., read, write, execute privileges) information for ECS resources.

Outputs

A history log file.

Success Criteria

Validation by the history log file that the adds, changes, and deletes were properly made. User ID attempts to maintain directory are rejected.

Assumptions and Constraints

With respect to requirement C-CSS-21100, only connect level authentication is being utilized in Ir1. Request and packet level authentication will be utilized in later releases. Only login failure notification is implemented in Ir1, so the other intrusions (unauthorized access to ECS resources, break-ins, and viruses and worms) noted in requirement C-MSS-70700 will not be tested at this time.

4.1.7 Test Case 1.7: Security Privilege Test (TC003.006)

This test case demonstrates system level Unix privilege integrity.

Test Inputs

A set of three valid Unix IDs and passwords with different system privileges.

Test Steps

Log on and log off using three valid ID/Password combinations.

For each ID/Password combination, use privileges allowed (file access, directory update, operator functions, etc.).

For each ID/Password combination, use privileges not allowed (file access, directory update, operator functions, etc.).

Test Outputs

Screen outputs showing the success or failure of the use of system privilege. System Management logs.

Success Criteria

Valid privilege use allowed. Invalid privilege use disallowed. System log records showing invalid attempts.

Assumptions and Constraints

None.

4.1.8 Test Case 1.8: Server Authentication (TC003.007)

This test case demonstrates a successful DCE server logon capability and the ability to interface with the DCE authentication database.

Test Inputs

DCE cell with an account in the DCE registry.

Test Steps

Log on and log off using three valid DCE machines.

Gathering information using a network analyzer, verify that the account name is ***hosts/<machine name>/self*.**

Test Outputs

Screen outputs showing the success of the server logon attempts. Response times of each logon and logoff event.

Success Criteria

Three successful server logon attempts, with each event occurring in under 15 seconds.

Assumptions and Constraints

The three servers specified will have three separate account names.

4.1.9 Test Case 1.9: Authentication Expiration (TC003.008)

This test case verifies that DCE authentication tickets granted to users and processes expire in the configured time.

Test Inputs

A registry database with a set of process and user privileges.

A set of valid DCE user IDs and passwords.

Test Steps

Set ticket expiration time parameter to a short period of time.

Log DCE users on and exercise valid user system privileges.

Wait until ticket expiration time expires.

Repeat user and process privilege actions.

Modify ticket expiration time and repeat.

Test Outputs

System logs showing failed privilege attempts.

Success Criteria

Privilege allowed prior to ticket expiration. Privilege disallowed after ticket expiration.

Assumptions and Constraints

None.

4.1.10 Test Case 1.10: Local Logons (rlogin) - Valid and Invalid (B01.01.01)

This test verifies that once connection to the system (H1) is established, a user is able to log onto another local host (H2), via basic LAN capabilities. All activity for each account is recorded in the history log file.

Test Inputs

Valid account names/passwords for H1 and H2. Invalid account names for H2. Valid account names but invalid passwords for H2.

Test Steps

Connection to H1 for valid account.

Connection to H2 for valid account, via rlogin from H1.

Exit H2.

Attempt reconnection to H2, via rlogin, for accounts with invalid account names or passwords.

View messages indicating incorrect rlogon to H2 for accounts with invalid account names or passwords.

Verify all activity is recorded in the History Log file.

Test Outputs

Connection to H2 is established for valid accounts, while connection to H2 is refused for invalid accounts. All activity by each account is recorded in the history log file, which is verified by the tester.

Success Criteria

The test will be deemed successful once the connections to H2 for the valid accounts are established and invalid attempts are denied.

Assumptions and Constraints

None.

4.1.11 Test Case 1.11: Remote Logons (Telnet H1-H2-H3) - Valid and Invalid (B01.01.02)

This test verifies that once connection to the system (H1), or to any other local host (H2), is established, a tester is able to log on to a remote host (H3), via basic WAN capabilities. All activity for the account is recorded in the history log file.

Test Inputs

Valid account names/passwords for H1, H2, and H3. Invalid account names for H3. Valid account names but invalid passwords for H3.

Test Steps

Connection to H1 for valid account.

Connection to H2 for valid account, via rlogin from H1.

Connection to H3 for valid account, via telnet from H2.

Exit H3.

Attempt reconnection to H3, via telnet, for accounts with invalid account names or passwords.

View messages indicating incorrect telnet to H3 for accounts with invalid account names or passwords.

Verify all activity is recorded in the History Log file.

Test Outputs

Connection to H3 is established for valid accounts, while connection to H3 is refused for invalid accounts. The remote login to H2 from H1 should remain connected. All activity by each account is recorded in the history log file, which is verified by the tester.

Success Criteria

The test will be deemed successful once the connections to H3 for the valid accounts are established and invalid attempts are denied.

Assumptions and Constraints

None.

4.1.12 Test Case 1.12: Remote Logons (Telnet H1-H3-H1) - Valid and Invalid (B01.01.03)

This test verifies that once a connection is made to a remote host (H3) from the local host (H1), a user is able to log back into the local host (H1), via basic WAN capabilities. This test verifies that connection into the LAN (NSI) from the WAN is possible.

Test Input

Valid account names/passwords for H1 and H3. Invalid account names for H1. Valid account names but invalid passwords for H1.

Test Steps

Connection to H1 for valid account.

Connection to H3 for valid account, via telnet from H1.

Exit H1 from H3.

Attempt reconnection to H1, via telnet, for accounts with invalid account names or passwords.

View messages indicating incorrect telnet to H1 for accounts with invalid account names or passwords.

Verify all activity is recorded in the History Log file.

Test Outputs

Connection to H1 is established for valid accounts, while connection to H1 is refused for invalid accounts. The telnet to H3 from H1 should remain connected. All activity by each account is recorded in the history log file, which is verified by the tester.

Success Criteria

The test will be deemed successful once the connections to H1, via H3, for the valid accounts are established and invalid attempts are denied.

Assumptions and Constraints

None.

4.1.13 Test Case 1.13: Syntax and Commands Simplification (TC011.001)

The purpose of Syntax and Commands Simplification is to verify that the complex DCE syntax and commands have been incorporated into accessible objects.

Test Inputs

Inputs to this test case include use of the OODCE class libraries.

Test Steps

Verify that the OODCE class libraries exist.

Call the libraries.

Verify that the available objects are accessible.

Test Outputs

The expected results of this test include successful use of the OODCE class libraries.

Success Criteria

This test will be deemed successful when all of the OODCE class libraries have been accessed and the objects have been executed.

Assumptions and Constraints

None.

4.1.14 Test Case 1.14: Sample Object Implementation (TC011.002)

This test case demonstrates, using a client/server architecture, the ability to demonstrate an object created using OODCE code and then link the object to C++ bindings.

<u>Test Inputs</u>

Inputs to this test case include use of the OODCE class libraries.

<u>Test Steps</u>

Demonstrate an object that was created using OODCE class libraries.

Verify that the object can bind using C++.

<u>Test Outputs</u>

The expected results of this test include being able to pass objects using OODCE.

<u>Success Criteria</u>

This test will be deemed successful when all objects developed using OODCE can bind to the client/server architecture.

<u>Assumptions and Constraints</u>

None

4.1.15 Test Case 1.15: Logoffs - Normal (B01.02.01)

This test verifies that when a tester, using a valid account, logs off a system or a host, the connection is properly closed to the system or the host.

<u>Test Inputs</u>

Valid account names/passwords for H1, H2, and H3.

<u>Test Steps</u>

Valid user connects to the system on Host 1, rlogs onto a local host (Host 2), then telnets onto a remote host (Host 3).

Log off the remote host (Host 3), then the local host (Host 2), and finally Host 1.

In each instance, another tester, using another valid account, is monitoring the system on each host to verify that connection to the host has been closed for the initial user.

Verify that all logon and logoff activity is recorded in the history log file.

Test Outputs

Monitor data for the activity of initial user, history log file records that have corresponding logon and logoff records.

Success Criteria

Verify that connection is established for initial user on the different hosts upon login and terminated upon logout. History log files should have corresponding logon and logoff records.

Assumptions and Constraints

None.

4.1.16 Test Case 1.16: Logoffs - Abnormal (B01.02.02)

This test verifies that when an abnormal event occurs and disconnects an account from the system (i.e., the tester's workstation is turned off, one of the hosts is powered off, a UNIX "kill", etc.), the system properly closes connection to the account. History log file entries are recorded for all activity. This test is to be repeated for the different types of abnormal events that will cause a disconnect from the system.

Test Inputs

Valid account names/passwords for H1, H2, and H3.

Test Steps

Log onto H1.

Perform rlogin to Host 2.

Perform telnet to Host 3.

Turn off (power down) H3.

Perform a UNIX kill command to terminate H2 connection.

Verify history log file records.

Test Outputs

History log file records. Port connections to the hosts are properly closed.

Success Criteria

History log file records which include a record that indicates that the H3 connection was closed and the H2 connection was terminated. If one of the hosts was powered off, then the user should be able to log back on once the host is powered back up.

Assumptions and Constraints

None.

4.1.17 Test Case 1.17: Login to EDF (T01-02.02.01)

This test verifies that a user is able to successfully access a host machine within the EDF from an external location.

Test Inputs

Valid account name/password for tester on external host (H1) and host within EDF (H2).

Test Steps

Perform standard remote login on EDF host machine from external host.

Verify history log file.

Test Outputs

Successful logon. History log file updated with tester activities.

Success Criteria

Connection to the system (H2) is established and the main screen of the host is displayed to the tester. The history log file is updated with the tester's logon activities.

Assumptions and Constraints

None.

4.1.18 Test Case 1.18: External Interfaces Integration Test (BC002.001)

This test case primarily verify that the existence of External interfaces implemented in Ir1, using the Gateway concept (refer to Figures B-1 and B-2).

Test Inputs

A series of Authentication Requests with valid ID and password, valid ID and unmatched password.

Test Steps

See Section 5

Test Outputs

Responses indicating acceptance of the Authentication Requests are logged.

Responses indicating rejection of the Authentication Requests are logged.

Success Criteria

A communiations connection is successfully established and all Authentication Requests sent by the clients are received and correctly determined to have valid IDs, valid/invalid password. Each Authentication Request is answered by a response indicating the acceptance or rejection of the request.

Assumptions and Constraints

A test driver will be used.

Assumptions and Constraints

The detailed external interfaces, accessed via the Gateway, can be seen in the Ir1 External Interface diagrams (one for the Data Server and one for Ingest), and they are covered by test cases 10.3 through 10.6, 11.3, and 11.4. The purpose of this test case is to demonstrate access to this interface

4.2 Messaging and File Transfer Thread Test

This thread demonstrates the messaging and file transfer functionality of Ir1. This thread will demonstrate the ECS message transfer capability across the internet, via E-Mail services, to designated user workstations. Other internet utilities will also be verified, as will the ftp file transfer methodology, as it applies to Ir1.

Special resources required for this thread test include:

- XRunner
- HP OpenView
- Network analyzer

4.2.1 Test Case 2.1: Internet Utilities Test (TS002.014)

Description

This test verifies the capability to browse WWW, to read from and post USENET newsgroups, read and send electronic mail via the Internet and provide access to a gopher and WAIS clients.

Test Inputs

Inputs to this test include URL locations and email messages.

Test Steps

See Section 5

Test Outputs

Outputs to this test include Email messages and URL locations.

Success Criteria

This test is successful if the ability to receive and distribute news, email, and WWW URLs over the Internet is demonstrated.

Assumptions and Constraints

None.

4.2.2 Test Case 2.2: Bulletin Board (BC012.004)

The purpose of the Bulletin Board test is to demonstrate the capability of interactive functionality for any bulletin board users. The bulletin board will be tested for the capability of users to either subscribe or unsubscribe to any bulletin board and to select a subscribed bulletin board for viewing all messages in that bulletin board. Capability of the bulletin board to respond to a message by sending the response to the bulletin board and/or to the author and/or any other user specified destination, will be tested.

The following search capability will be tested for bulletin board users:

a. search for a string in message headers and in message text

b. search by author

c. search by subject

A catch-up feature which excludes user specified messages from appearing in the bulletin board when it is viewed next time, will be tested. The bulletin board services must demonstrate that users can post messages to bulletin board(s), maintain an access history for each bulletin board per user basis, which tracks the messages read by the user for each bulletin board. Saving messages to the local bulletin board system will be tested, along with attaching ASCII or binary files to a message being sent. Also, a means to retrieve files will be tested.

The BBS must demonstrate the following bulletin board configuration options:

a. screen size

b. number of messages displayed on a screen

c. screen colors (background/foreground)

d. read message indicator

Test Inputs

Sequence of interactive commands simulating BBS usage for search, configuration, message retrieval, etc.

Test Steps

Verify that the BBS is based on TCP/IP, NNTP, SMTP and USENET message standard.

Demonstrate the functionality of the BBS to support multiple bulletin boards.

Verify that multiple users (registered and non-registered) can be accessed concurrently.

Verify that each bulletin board can contain multiple messages.

Verify, as an M&O user, the ability to: create the new bulletin board, delete existing bulletin boards, delete messages from a bulletin board, back up bulletin boards, force users off a bulletin board or the entire bulletin board service for backup, collect history and or statistical information and back up bulletin boards.

Send a message to bulletin boards.

Copy or save a posted message to their local system.

Verify the ability to respond to a posted message by sending the response message to: the bulletin board, author of the original message and named destinations.

Demonstrate the use of the Bulletin Board via interactive mode from the command line.

Demonstrate the ability of the CSS Bulletin Board Service to allow users to subscribe and/or unsubscribe to a bulletin board.

Demonstrate the capability to subscribe to a bulletin board and view a summary for each message in it.

User will respond to a message by sending the response to the bulletin board, author of the message and/or any other operator specified destination.

Using the search capability provided by the BBS: search for a string, search by author or search by subject.

Verify that a catch-up feature which excludes user specified messages from appearing in the bulletin board next time it is viewed.

Test Outputs

Outputs include: access of multiple bulletin boards with multiple messages posted to each, new bulletin board, deleted bulletin boards, back ups, access history and statistical information, responses to messages, subscribed to bulletin boards, unsubscribed bulletin boards, and copy or saves of a message to their local system.

Success Criteria

When all validation and verification of all BBS functions have been determined to have satisfied the overall test requirements.

Assumptions and Constraints

None.

4.2.3 Test Case 2.3: E-Mail (TC006.002)

The E-Mail test will demonstrate the capability of the messaging service to manage and interact with user E-mail. This mailbox will be tested for, copying and/or moving messages from the MAILBOX to the user defined folders, and for providing an access control feature which requires authentication for access to the Mail tool via login and password. The MAILBOX will also be tested for allowing users to set an automatic time period for deletion of messages to help manage the MAILBOX size, by removing old messages after confirmation. The message editor will be tested for the capability of composing messages, by providing a title/subject field for the message and various destinations.

Test Inputs

Inputs require creating a user defined MAILBOX which will store incoming messages in the mailbox folders, created for long term archiving. Summary status for all mail messages will be verified. In addition, a reply to a message, composing a message over the editor, and sending a message, (by creating a private mailing list and creating a public mailing list) will be required.

Test Steps

Demonstrate the capability to access the electronic mail service in interactive mode.

Reply, to the author and to all destinations addressed in the incoming message.

Verify that a "MAILBOX" is provided, where the incoming messages are stored.

Demonstrate the availability of operator defined folders for archiving and verify that you can copy/move messages from the "MAILBOX" to specified folders.

Retrieve a summary of all messages, containing at a minimum title/subject and name of author.

Verify that an editor is provided to compose a message and that there is a provided title/subject field for a message.

Send the message to multiple destinations (repeat for: a single user, an operator manned position, a mailing list, and a site which consists of several operators).

Create a private mailing list, verify that the lists can contain multiple destinations for individual operators.

Create a public mailing list, verify that the list contains multiple destinations accessible to all operators.

Test Outputs

The expected results of these tests are a successful demonstration of all the required interactive user functions provided by the MAILBOX and folder. Validation and verification of each test must be complete and satisfactory. A reply to a message, an archived message, summary of all messages, private and public mailing lists.

Success Criteria

When all validation and verification of all Mailtool functions have been determined to have satisfied the overall test requirements.

Assumptions and Constraints

None.

4.2.4 Test Case 2.4: EDF to DAAC Message Transfer (TC010.001)

The EDF to DAAC message transfer test will demonstrate E-mail message transfer capability, TO and FROM the EDF workstation. This test will include, *single* message transfers TO and FROM a single DAAC, and a *distributed* message transfer FROM EDF TO all DAACs.

<u>Test Inputs</u>

Inputs include, creating a simple mail message to transfer across the internet from the EDF workstation and receiving a mail message from the internet DAACs on the EDF workstation.

<u>Test Steps</u>

Single message transfer

Login on EDF workstation

Mail message to each DAAC individually

Check for successful transfer confirmation

Distributed message transfer

Login on EDF workstation

Mail message to each DAAC simultaneously

Check for successful transfer confirmation

Receive message transfer from remote DAAC workstation

Transfer mail message from a remote DAAC workstation to EDF workstation

Login on EDF workstation

Enter: mail

Verify new message received in mailbox

<u>Test Outputs</u>

The expected results of this test include successful message transfers TO and FROM the designated Ir1 workstations, as they're defined in each test.

<u>Success Criteria</u>

When a validation and verification of all messages transferred or received have been determined successful TO and FROM the designated workstations.

<u>Assumptions and Constraints</u>

All workstations are configured to transfer and receive mail messages via E-mail.

4.2.5 Test Case 2.5: E-Mail from EDF to GSFC (T01-02.05.01)

This test verifies that a tester using valid accounts, via basic LAN and WAN capabilities, is able to send e-mail messages from an account at EDF to an account at GSFC.

Test Inputs

Valid account names/passwords for accounts at both EDF and GSFC. Tester sends e-mail message from EDF account at GSFC.

Test Steps

Connection to the respective hosts, message received by account at GSFC.

Test Outputs

Tester receives e-mail message from EDF at GSFC. History log file will record all activities and transactions.

Success Criteria

The email from EDF to GSFC is transmitted successfully.

Assumptions and Constraints

None.

4.2.6 Test Case 2.6: E-Mail from EDF to LaRC (T01-02.05.02)

This test verifies that a tester using valid accounts, via basic LAN and WAN capabilities, is able to send e-mail messages from an account at EDF to an account at LaRC.

Test Inputs

Valid account names/passwords for accounts at both EDF and LaRC. Tester sends e-mail message from EDF account at LaRC.

Test Steps

Connection to the respective hosts, message received by account at LaRC.

Test Outputs

Tester receives e-mail message from EDF at LaRC. History log file will record all activities and transactions.

Success Criteria

The email transmission from EDF to LaRC completes successfully.

Assumptions and Constraints

None.

4.2.7 Test Case 2.7: SCF to DAAC Message Transfer (TC010.002)

The SCF to DAAC message transfer test will demonstrate E-mail message transfer capability, TO and FROM the SCF workstation. This test will include, *single* message transfers TO and FROM a single DAAC, and a *distributed* message transfer FROM SCF TO all DAACs (may be emulated).

Test Inputs

Inputs include, creating a simple mail message to transfer across the internet from the SCF workstation and receiving a mail message from the internet DAACs on the SCF workstation.

Test Steps

Single message transfer

Login on SCF workstation

Mail message to each DAAC individually

Check for successful transfer confirmation

Distributed message transfer

Login on SCF workstation

Mail message to each DAAC simultaneously

Check for successful transfer confirmation

Receive message transfer from remote DAAC workstation

Transfer mail message from a remote DAAC workstation to SCF workstation

Login on SCF workstation

Enter: mail

Verify new message received in mailbox

Test Outputs

The expected results of this test include successful message transfers TO and FROM the designated Ir1 workstations, as they're defined in each test.

Success Criteria

When a validation and verification of all messages transferred or received have been determined successful TO and FROM the designated workstations.

Assumptions and Constraints

All workstations are configured to transfer and receive mail messages via E-mail. If available for use may use real SCF. If an SCF is not available we will emulate the functionality.

4.2.8 Test Case 2.8: E-Mail from EDF to EDC (T01-02.05.04)

This test verifies that a tester using valid accounts, via basic LAN and WAN capabilities, is able to send e-mail messages from an account at EDF to an account at EDC.

Test Input

Valid account names/passwords for accounts at both EDF and EDC. Tester sends e-mail message from EDF account at EDC.

Test Steps

Connection to the respective hosts, message received by account at EDC.

Test Outputs

Tester receives e-mail message from EDF at EDC. History log file will record all activities and transactions.

.Success Criteria

The e-mail message from EDF to EDC completes successfully.

Assumptions and Constraints

None.

4.2.9 Test Case 2.9: DAAC to DAAC Message Transfer (TC010.003)

The DAAC to DAAC message transfer test will demonstrate E-mail message transfer capability, TO and FROM the DAAC workstations. This test will include, *single* message transfers TO and FROM a single DAAC, and a *distributed* message transfer TO and FROM all DAACs.

Test Inputs

Inputs include, creating a simple mail message to transfer across the internet from one DAAC workstation and receiving a mail message from the internet DAACs on the same DAAC workstation.

Test Steps

Single message transfer

Login on DAAC workstation

Mail message to each DAAC individually

Check for successful transfer confirmation

Distributed message transfer

Login on DAAC workstation

Mail message to each DAAC simultaneously

Check for successful transfer confirmation

Receive message transfer from remote DAAC workstation

Transfer mail message from a remote DAAC workstation to DAAC workstation

Login on DAAC workstation

Enter: mail

Verify new message received in mailbox

Test Outputs

Successful message transfer confirmation and the verification

Success Criteria

All message transfer completes successfully.

Assumptions and Constraints

None.

4.2.10 Test Case 2.10: E-Mail from DAAC to DAAC (T01-02.05.05)

This test verifies that a tester, using valid accounts, is able to send e-mail between DAAC sites. This test will be completed at each DAAC site.

Test Input

Valid account names/passwords for accounts at both DAACs. Tester sends e-mail message from site to site.

Test Steps

Connection to the respective hosts, message received by account at DAAC 2.

Test Outputs

Tester receives e-mail message from DAAC 1 at DAAC 2. History log file will record all activities and transactions.

Success Criteria

The account at DAAC 2 connects successfully.

Assumptions and Constraints

None.

4.2.11 Test Case 2.11: E-Mail- Asynchronous Messaging (TC006.001)

The purpose of the E-mail test case is to demonstrate the availability of asynchronous messaging (loosely coupled with a single message queue) through the use of the E-Mail API.

Test Inputs

Inputs to this test case include a series of e-mail transmissions using API calls via an Ir1 application.

Test Steps

Activate test script that calls the e-mail API and sends mail messages. Activate an Ir1 application that calls the e-mail API and sends mail messages.

Test Outputs

Outputs include receipt of valid E-mail message and notification of message not delivered. Mail message sent via the SMTP and X.400 protocols.

Success Criteria

This test will be deemed successful when all e-mail messages are successfully transmitted to their designated destinations.

Assumptions and Constraints

None.

4.2.12 Test Case 2.12: Sending E-Mail Messages to Local and Remote Hosts (B01.05.01)

This test verifies that a tester using a valid account, via basic LAN and WAN capabilities, is able to e-mail messages to other accounts on the same host (H1), on a local host (H2), or on a remote (H3) host. In all cases, records of the transactions are recorded in the history log file.

Test Input

Valid account names/passwords for account A on local H1, account B on local H1, account C on local H2, and account D on remote H3. Message 1 sent from account A to account B. Message 2 sent from account A to account C. Message 3 sent from account A to account D.

Test Steps

Connection to the respective hosts, message 1 received by account B, message 2 received by account C, and message 3 received by account D.

History log file records of all activity and transactions by the tester.

Test Outputs

The history log will record the logon to the system by each account, it will record the transmission of the e-mail messages, it will record the resource usage, response time, and the number of transactions.

Success Criteria

The connection between accounts will successfully connect.

Assumptions and Constraints

None.

4.2.13 Test Case 2.13: Receiving E-Mail Messages from Local and Remote Hosts (B01.05.02)

This test verifies that a tester using a valid account, via basic LAN and WAN capabilities, is able to receive e-mail messages from the same host (H1), a local host (H2), or a remote host (H3). Each message, in this case, will be sent simultaneously to the same account (account A). In all cases, records of the transactions are recorded in the history log file.

Test Input

Valid account names/passwords for account A on local H1, account B on local H1, account C on local H2, and account D on remote H3. Messages from accounts B, C, and D to account A.

Test Steps

Connection to the respective hosts, messages received by account A.

History log file records of all activity and transactions by the tester.

Test Outputs

The history log will record the logon to the system by each account, it will record the transmission of the e-mail messages, it will record the resource usage, response time, and the number of transactions.

Success Criteria

The connection to the respective hosts will be successful.

Assumptions and Constraints

None.

4.2.14 Test Case 2.14: Client/User File Transfer (TC009.001)

The purpose of the Client/User File Transfer test is to demonstrate communications among the appropriate DAAC workstations that are available for Ir1. Demonstrate the ability of a valid Ir1 user to ftp (put and get) a file to a workstation that is not in the Ir1 Cell. Verify that a valid Ir1 user can transfer files using the interactive transfer mode. Demonstrate the functionality of ftp commands (mput, mget, open, close, cd, ascii, binary, chmod, delete, dir, hash, image, prompt, system, type, etc.)

Test Inputs

Inputs to this test case include a series of file transfers using ftp for various data files. The data files will be consistent, in size and content, with actual Ir1 data files.

Test Steps

Verify that there are test accounts setup on all of the Ir1 implemented DAAC workstations (EDF - edf-bb, epserver, GSFC - ecsgsfc1, ecs-global, MSFC - hydra, meteor, LaRC - ecs, nephos and EDC - ecs-hp1, ecs-alpha1).

Demonstrate that each of the workstations can transfer files to other workstations and receive files that have been transferred from other workstations.

Verify that the transfers were completed in a timely and efficient manner. Efficiency is determined not only by speed but also by the accuracy in which a file reaches its destination.

Demonstrate functionality of ftp commands.

Test Outputs

The expected results of this test include successful file transfers (put and get) from each of the Ir1 workstations. Screen outputs of expected results for each ftp command.

Success Criteria

This test will be deemed successful when all files are transferred from their designated source to their designated destination and are verified to match the expected file. All ftp commands function as expected.

Assumptions and Constraints

None.

4.2.15 Test Case 2.15: Transmit File from EDF to GSFC (ftp) (T01-02.04.01)

This test verifies that a tester, using a valid account, is able to transmit a file from EDF to an account on host machine at GSFC. This test verifies the connectivity of the EDF LAN to the system WAN.

Test Inputs

Valid login for tester on H1 at EDF and H2 at GSFC. Data file for file transfer (ftp) from H1 to H2.

Test Steps

ftp completes data file transfer from H1 to H2.

H2 directory listings verify that data file was transferred.

Test Outputs

The history log file will record the logon to both hosts, it will record the transmission of the ftp transaction, it will record the resource usage, response time, and the number of transactions. Checksum of data files on H1 and H2 should equal.

Success Criteria

H1 and H2 will successfully transfer a data file.

Assumptions and Constraints

None.

4.2.16 Test Case 2.16: Transmit File from EDF to LaRC (ftp) (T01-02.04.02)

This test verifies that a tester, using a valid account, is able to transmit a file from EDF to an account on host machine at LaRC. This test verifies the connectivity of the EDF LAN to the system WAN.

Test Inputs

Valid login for tester on H1 at EDF and H2 at LaRC. Data file for file transfer (ftp) from H1 to H2.

Test Steps

ftp completes data file transfer from H1 to H2.

H2 directory listings verify that data file was transferred.

Test Outputs

The history log file will record the logon to both hosts, it will record the transmission of the ftp transaction, it will record the resource usage, response time, and the number of transactions. Checksum of data file on H1 and H2 should equal.

Success Criteria

The data file will successfully transmit between H1 and H2.

Assumptions and Constraints

None.

4.2.17 Test Case 2.17: Transmit File from EDF to MSFC (ftp) (T01-02.04.03)

This test verifies that a tester, using a valid account, is able to transmit a file from EDF to an account on host machine at MSFC. This test verifies the connectivity of the EDF LAN to the system WAN.

Test Inputs

Valid login for tester on H1 at EDF and H2 at MSFC. Data file for file transfer (ftp) from H1 to H2.

Test Steps

ftp completes data file transfer from H1 to H2.

H2 directory listings verify that data file was transferred.

Test Outputs

The history log file will record the logon to both hosts, it will record the transmission of the ftp transaction, it will record the resource usage, response time, and the number of transactions. Checksum of data file on H1 and H2 should equal.

Success Criteria

The ftp data file transfer will complete between H1 and H2.

Assumptions and Constraints

None.

4.2.18 Test Case 2.18: Transmit File from EDF to EDC (ftp) (T01-02.04.04)

This test verifies that a tester, using a valid account, is able to transmit a file from EDF to an account on host machine at EDC. This test verifies the connectivity of the EDF LAN to the system WAN.

Test Inputs

Valid login for tester on H1 at EDF and H2 at EDC. Data file for file transfer (ftp) from H1 to H2.

Test Steps

ftp completes data file transfer from H1 to H2.

H2 directory listings verify that data file was transferred.

Test Outputs

The history log file will record the logon to both hosts, it will record the transmission of the ftp transaction, it will record the resource usage, response time, and the number of transactions. Checksum of data file on H1 and H2 should equal.

Success Criteria

The data transmission from H1 to H2 completes successfully.

Assumption

None.

4.2.19 Test Case 2.19: Transmit File from EDF to GSFC (rcp) (T01-02.04.05)

This test verifies that a tester, using a valid account, is able to transmit a data file to an account on a machine at GSFC via remote file copy (rcp).

Test Inputs

Valid login for tester on H1 at EDF and H2 at GSFC. Data file for remote file copy (rcp) from H1 to H2.

Test Steps

rcp completes data file transfer from H1 to H2.

H2 directory listings verify that data file was transferred.

Test Outputs

The history log file record the logon onto both hosts, it will record the transmission of the rcp transaction, it will record the resource usage, response time, and the number of transactions. Checksum of data files on H1 and H2 should equal.

Success Criteria

The rcp completes successfully between H1 and H2.

Assumptions and Constraints

None.

4.2.20 Test Case 2.20: Transmit File from EDF to LaRC (rcp) (T01-02.04.06)

This test verifies that a tester, using a valid account, is able to transmit a data file to an account on a machine at LaRC via remote file copy (rcp).

Test Inputs

Valid login for tester on H1 at EDF and H2 at LaRC. Data file for remote file copy (rcp) from H1 to H2.

Test Steps

rcp completes data file transfer from H1 to H2.

H2 directory listings verify that data file was transferred.

Test Outputs

The history log file record the logon onto both hosts, it will record the transmission of the rcp transaction, it will record the resource usage, response time, and the number of transactions. Checksum of data files on H1 and H2 should equal.

Success Criteria

The rcp completes successfully between H1 and H2.

Assumptions and Constraints

None.

4.2.21 Test Case 2.21: Transmit File from EDF to MSFC (rcp) (T01-02.04.07)

This test verifies that a tester, using a valid account, is able to transmit a data file to an account on a machine at MSFC via remote file copy (rcp).

Test Inputs

Valid login for tester on H1 at EDF and H2 at MSFC. Data file for remote file copy (rcp) from H1 to H2.

Test Steps

rcp completes data file transfer from H1 to H2.

H2 directory listings verify that data file was transferred.

Test Outputs

The history log file record the logon onto both hosts, it will record the transmission of the rcp transaction, it will record the resource usage, response time, and the number of transactions. Checksum of data files on H1 and H2 should equal.

Success Criteria

The rcp completes successfully between H1 and H2.

Assumptions and Constraints

None.

4.2.22 Test Case 2.22: Transmit File from EDF to EDC (rcp) (T01-02.04.08)

This test verifies that a tester, using a valid account, is able to transmit a data file to an account on a machine at EDC via remote file copy (rcp).

Test Inputs

Valid login for tester on H1 at EDF and H2 at EDC. Data file for remote file copy (rcp) from H1 to H2.

Test Steps

rcp completes data file transfer from H1 to H2.

H2 directory listings verify that data file was transferred.

Test Outputs

The history log file record the logon onto both hosts, it will record the transmission of the rcp transaction, it will record the resource usage, response time, and the number of transactions. Checksum of data files on H1 and H2 should equal.

Success Criteria

The rcp completes successfully between H1 and H2.

Assumptions and Constraints

None.

4.2.23 Test Case 2.23: Anonymous ftp (TC009.003)

The purpose of the Anonymous ftp test is to demonstrate the capability of non Ir1 users transferring or retrieving data from the DAAC workstations. There is a special directory (/users/ftp/pub) setup for this functionality to ensure that the DAAC workstations can not be violated by providing outside users access to the system. Verify that non Ir1 users can retrieve or transfer files to this designated directory. Verify that Ir1 users can post files to the directory to be retrieved from non Ir1 users or retrieve the files that the non Ir1 users have posted.

Test Inputs

Inputs to the test case include a series of file transfers (put and get) using anonymous ftp for various file sizes.

Test Steps

Inspect each of the DAAC Ir1 workstations to ensure that it has a /users/ftp/pub directory with read/write privileges for the world.

Verify that the user is unable to access any other directories within the cell.

Place a file in this directory and verify that a non ECS user can retrieve the file by using anonymous ftp.

Verify that a non ECS user can copy a file to the same directory using anonymous ftp.

Demonstrate that a valid Ir1 user can then retrieve the file that was previously transferred to the /users/ftp/pub directory.

Test Outputs

The expected results of this test include a report indicating that the file was successfully transferred or retrieved using anonymous ftp.

Success Criteria

This test will be deemed successful when non ECS users are able to retrieve files from or transfer files to the /users/ftp/pub directory on a DAAC workstation. The files that have been transferred to this directory from non ECS users will be moved to the appropriate locations by valid Ir1 users.

Assumptions and Constraints

None.

4.2.24 Test Case 2.24: Application File Transfer (TC009.004)

The purpose of the Application File Transfer test is to demonstrate the ability of the application to transfer data files using API calls.

Test Inputs

Inputs to this test case include a series of file transfers using API calls for various data files, by calling an Ir1 application.

Test Steps

Verify that there are test accounts setup on all of the Ir1 implemented DAAC workstations (EDF - edf-bb, epserver, GSFC - ecsgsfc1, ecs-global, MSFC - hydra, meteor, LaRC - ecs, nephos).

Using edf-bb as the server, create a data file to initialize the transfers of various file sizes to each of the Ir1 workstations (client).

This transfer of a data file will be launched throughout the day. Both receives and puts will be executed.

Analyze the data.

Test Outputs

The expected results of this test include successful file transfers from each of the Ir1 workstations. Reports will show where deviations occur.

Success Criteria

This test will be deemed successful when all files are transferred from their designated source to their designated destination, and are verified to match the expected file.

Assumptions and Constraints

None.

4.2.25 Test Case 2.25: Network Filtering Test (BC002.003)

This test illustrates the internetworking filtering of packets on port/socket and source and/or destination address. V0 network management configuration file will be analyzed in order to illustrate these capabilities.

Test Inputs

Router configuration file contains the filtering conditions, network packets with combinations of port/socket identification and source/destination addresses.

Test Steps

See Section 5.

Test Outputs

The expected results of this test include successful inspect of the V0 network Router configuration file and router reference book.

Success Criteria

This test will be deemed successful when the necessary document about router used in V0 network has been inspected and the V0 network router configuration file has been inspected.

Assumptions and Constraints

No actul test steps will be performed because of the ownership.

4.2.26 Test Case 2.26: Multiple Accounts Transmitting Large Data Files to GSFC DAAC (B01.07.01)

This test verifies that multiple accounts are able to transmit large data files over the WAN to GSFC.

Test Inputs

Accounts A through F are logged on with valid account names/passwords. Accounts A through F will simultaneously transmit a large data file to GSFC, using ftp the "hash" option, to valid accounts at GSFC.

Test Steps

Accounts A through F will view their perspective screens and notice a slight delay in the rate the hashing appears.

Test Outputs

The history log will record each input that was initiated by each account. The system response will slow down due to the network traffic from the multiple ftp transactions.

Success Criteria

The multiple ftp transactions will complete successfully and the system slow down due to the network traffic.

Assumptions and Constraints

None.

4.2.27 Test Case 2.27: Multiple Accounts Transmitting Large Data Files Within the EDF (B01.07.02)

This test verifies that multiple accounts can transmit large data files over the Ethernet LAN within the EDF.

Test Inputs

Accounts A through G are logged on with valid account names/passwords. Accounts A through F will simultaneously transmit a large data file within the EDF to account G using ftp with the "hash" option.

Test Steps

Accounts A through F will view their perspective screens and notice a slight delay in the rate the hashing appears.

Test Outputs

The history log will record each input that was initiated by each account. The system response will slow down due to the network traffic from the multiple ftp transactions.

Success Criteria

The system response will slow down due to the network traffic from the multiple ftp transactions.

Assumptions and Constraints

None.

4.2.28 Test Case 2.28: Fault Notification sent via NSI (T01-02.05.07)

This test verifies that the NSI, NASA Science Internet, transmitting a fault notification from GSFC to EDF. The fault notification will be in the form of a consistently formatted electronic message that can be automatically parsed by a receiving program from ECS. It will contain enough information to determine the nature of the fault and which sites are affected.

Test Inputs

The NSI schedules preventive maintenance to one of its connections to GSFC. The fault notification will contain the preventive maintenance schedule as to when the maintenance begins, ends, and to what locations are affected. In this case, the EDF is affected.

Test Steps

See Section 5

Test Outputs

Fault notification sent electronically as an alert. Also, the fault notification contributes to an audit trail that assist with performing network analysis.

Success Criteria

The fault notification will reflect all of the pertinent information users need to know.

Assumptions and Constraints

None.

4.3 System Management Thread Test

The purpose of this thread is to verify the functionality of the Directory/Naming Service (used to uniquely associate a name with resources/principals, along with their attributes), the Distributed Time Service (DTS - used to synchronize time services across hosts located within the same DCE cell), and the HP OpenView Enterprise Management Framework (used to monitor software and hardware objects, as well as detect and display alarms).

Special resources required for this thread test include:

- Cell Directory Service Command Program (cdscp)
- XRunner
- LoadRunner
- Privileges to execute DTS management functions
- Sample science data files
- HP OpenView
- Network analyzer

4.3.1 Test Case 3.1: X/Open Functions (TC004.001)

This test case will demonstrate that the Directory/Naming Service can provide basic X/Open Federated Naming functions.

Test Inputs

Perform X/Open functions.

Test Steps

Log onto a DAAC workstation.

Create a list of directory entries.

Verify that the Directory Service determines which naming service to use from a given context.

List the names in the directory.

Get an entry from the directory.

Delete a directory.

Create attributes for a directory.

List the attributes for a directory.

Get attribute information for a directory.

Set attribute information for a directory.

Delete attribute information for a directory.

Associate a directory to attributes.

Test Outputs

The Directory/Naming Service should display all relative information pertaining to the above functions.

Success Criteria

The Directory/Naming Service should be able to perform all of the above functions.

Assumptions and Constraints

None.

4.3.2 Test Case 3.2: Replication (TC004.002)

This test case will demonstrate that the Directory/Naming Service can provide and maintain copies of the namespace across DAAC workstations. Read/write access and propagation of master changes to replicas will also be verified.

Test Inputs

cdscp commands

Test Steps

Log onto a DAAC server workstation.

Using cdscp commands, create a clearinghouse (master copy).

Log onto another server workstation in the same DAAC.

From the second workstation, create a read-only copy of the clearinghouse created in step 2.

Attempt to write to a file in the replica created in step 4. CDS should prohibit this action.

Ensure that the attribute for the propagation of changes is set to "low" (no immediate skulk).

Modify a file in the master clearinghouse. These changes should not be applied to the read-only replica.

Manually propagate the file changes in step 7. These changes should now be applied to the read-only replica.

Ensure that the attribute for the propagation of changes is now set to "high" (immediate skulk).

Modify another file in the master clearinghouse.

These modifications should now be automatically applied to the read-only replica.

Test Outputs

The Directory/Naming Service should display all relative information pertaining to the above functions.

Success Criteria

This test will be deemed successful when the Directory/Naming Service enables one to perform all of the above functions.

Assumptions and Constraints

It is assumed that the DAAC file system will be populated with a number of directories, sub directories, data files, and other objects.

4.3.3 Test Case 3.3: Distribution (TC004.003)

This test case will demonstrate that the Directory/Naming Service can distribute and manage namespaces and replicas across different hosts.

Test Inputs

cdscp commands

Test Steps

Using cdscp administration commands, partition a namespace in a DAAC among three different hosts.

Using cdscp administration commands, replicate partitions of this namespace across three different hosts.

Via the Global Data Service (GDS), attempt to access namespace information between two DAAC hosts.

Verify that the namespace can be updated automatically and manually.

Verify that the CSS Directory Service shall interact with the Security Service to provide host based security to the entries in the namespace.

Perform the cdscp command to denote the relative root of several namespaces.

Verify that the DCE profile function reduces search time for name lookups.

Verify that a local cache is maintained to keep recent lookup information.

Test Outputs

The Directory/Naming Service should display all information pertaining to each replica's location. An updated namespace and a local cache maintaining the most recent lookup information to be retrieved easily.

Success Criteria

This test will be deemed successful when the Directory/Naming Service enables one to perform all of the above functions.

Assumptions and Constraints

It is assumed that the DAAC file system will be populated with a number of directories, sub directories, data files, and other objects.

4.3.4 Test Case 3.4: Single Host Time Synchronization (TC005.001)

The purpose of the single host time synchronization test is to verify that time on a given client is synchronized (within the maximum inaccuracy set) to the UTC.

Test Inputs

Inputs to this test case include:

- DTS tuning characteristics values:

Parameter	Default Value
Maximum Inaccuracy	100 ms
Server Sync Hold Down	2 mins
Client Sync Hold Down	10 mins
Error Tolerance	10 mins
Local Set Timeout	2 secs
Global Set Timeout	15 secs
Query Attempts	3

Test Steps

Execute an XRunner script and cron job to simultaneously get the UTC as well as the DCE time from a given host.

Test Outputs

The expected results of this test include RPC entries with DTS time stamp.

Success Criteria

This test will be deemed successful when UTC time and local client DCE time are synchronized.

Assumptions and Constraints

In accordance with the DCE Admin guide, time stamps retrieved are cell specific. In other words, there is no time synchronization between DCE cells. All time synchronization tests will therefore be conducted within a single DCE cell (to verify time synchronization among the hosts within that cell). The focus will be on inaccuracy verification. The test case may require the verification for other parameters.

4.3.5 Test Case 3.5: Multiple Host Time Synchronization (TC005.002)

The purpose of the multiple host time synchronization test is to verify that time on multiple clients is synchronized (within the maximum inaccuracy set) to the UTC in a given DCE cell.

Test Inputs

Inputs to this test case include:

- DTS tuning characteristics values:

Parameter	Default Value
Maximum Inaccuracy	100 ms
Server Sync Hold Down	2 mins
Client Sync Hold Down	10 mins
Error Tolerance	10 mins
Local Set Timeout	2 secs
Global Set Timeout	15 secs
Query Attempts	3

Test Steps

Execute a LoadRunner script and cron job to simultaneously get the UTC from a given host and the DCE time from all other hosts.

Test Outputs

The expected results of this test include RPC entries with DTS time stamp.

Success Criteria

This test will be deemed successful if all time stamps across the DCE cell result in identical time identification within the acceptable DCE tolerances.

Assumptions and Constraints

In accordance with the DCE Admin guide, time stamps retrieved are cell specific. In other words, there is no time synchronization between DCE cells. All time synchronization tests will therefor be conducted within a single DCE cell (to verify time synchronization among the hosts within that cell).

4.3.6 Test Case 3.6: Inaccuracy Injection (TC005.003)

The purpose of the Inaccuracy Injection test is to demonstrate that DTS performs time synchronization between hosts in a DCE cell when drift rates and inaccuracies exceed the system-reported maximum clock drift rate and inaccuracy set by the administrator.

Test Inputs

Inputs to this test case include:

- DTS tuning characteristics values:

Parameter	Default Value
Maximum Inaccuracy	100 ms
Server Sync Hold Down	2 mins
Client Sync Hold Down	10 mins
Error Tolerance	10 mins
Local Set Timeout	2 secs
Global Set Timeout	15 secs
Query Attempts	3

Test Steps

Through the DTS management functions set the drift rate on a given host to exceed the maximum allowable rate.

Execute LoadRunner script from test case 2 to capture time stamps at intervals of 15 seconds.

Test Outputs

The expected results of this test include DTS time displays before and after inaccuracies are introduced.

Success Criteria

This test will be deemed successful when the correct time within tolerance range is computed by DTS and propagated to all hosts.

Assumptions and Constraints

At least two of the three DTSs are up and accurately running.

4.3.7 Test Case 3.7: DTS Management (TC005.004)

The purpose of the DTS Management test is to demonstrate that DTS DCE services are configurable within the DTS pyramid. Clerks, Servers (local, global, couriers, backup couriers, time providers), maximum drift rates and inaccuracies, and time representation formats will be modified as part of an M&O type scenario.

Test Inputs

Inputs to this test case include:

- DTS tuning characteristics values:

Parameter	Default Value
Maximum Inaccuracy	100 ms
Server Sync Hold Down	2 mins
Client Sync Hold Down	10 mins
Error Tolerance	10 mins
Local Set Timeout	2 secs
Global Set Timeout	15 secs
Query Attempts	3

- Commands for DTS modification

Test Steps

dtscp

dtsd

Clerk and server startup and shutdown

Displaying and setting the time

Installing new servers

Converting clerks to servers

Identifying and fixing faulty servers

System tuning

Test Outputs

The expected results of this test include screen displays for the DTS commands.

Success Criteria

This test will be deemed successful when DTS management functions are demonstrated.

Assumptions and Constraints

None.

4.3.8 Test Case 3.8: DTS Security (TC005.005)

The purpose of the DTS Security test is to demonstrate that security functions are invoked in the DCE cell.

Test Inputs

Inputs to this test case include:

- DTS tuning characteristics values:

Parameter	Default Value
Maximum Inaccuracy	100 ms
Server Sync Hold Down	2 mins
Client Sync Hold Down	10 mins
Error Tolerance	10 mins
Local Set Timeout	2 secs
Global Set Timeout	15 secs
Query Attempts	3

Test Steps

Exercise DTS privileges for:

- o Principal
- o Server Group

Test Outputs

The expected results of this test include screen displays for the DTS commands.

Success Criteria

This test will be deemed successful when DTS security functions are demonstrated.

Assumptions and Constraints

In accordance with the DCE Admin guide, time stamps retrieved are cell specific. In other words, there is no time synchronization between DCE cells. All time synchronization tests will therefore be conducted within a single DCE cell (to verify time synchronization among the hosts within that cell).

4.3.9 Test Case 3.9: DBMS Interface (TC013.003)

The purpose of the DBMS Interface test is to verify that an interface is provided to ingest the history log (flat file) into the database. Verify that the all of the desired information from the history log is ingest to the database (i.e., verify that each entries logged into the flat file contains all of the appropriate information: application start or stop time, application name/version, event message information, event message type, event disposition narrative, user principle information, and environment information). Verify that the PMAS provides the queries to generate performance reports. Note: CPU, memory, and disk utilization are recorded during the TC013.005 test.

Test Inputs

Inputs to this test case include history log files created from each of the available DAACs and from the EDF. Inputs to this history log file include a series of system logins, file transfers using

FTP and RPC for a given data file, activation, execution, termination of other application software provided as part of Ir1, and performance data. Generated Sybase SQL queries.

Test Steps

Tester will login to an Ir1 application using a valid id and valid password, an invalid id and valid password, a valid id and invalid password and an invalid id and invalid password.

Tester will implement the available applications (making sure for each application started, it will also be closed) to make entrants into the history log.

Tester will set a threshold and run a test that will surpass the threshold.

Verify that the history log will record the event via the CSS implemented API.

Simulate the retrieval of the science algorithm performance data from the local History Logs using the CSS provided APIs.

Tester will switch to the history log directory.

Ingest the history log into the database.

Verify the desired information from the history log is entered into the database.

Verify that the PMAS provides queries to generate performance statistics from the performance data stored in the database (queries will include single and multi-level indices).

Verify that network statistics for a configurable period of time for performance data store in the management database (average, median, maximum, minimum, ratios, rates and standard deviations).

Verify that the generated performance data is stored in the PMAS.

Print the M&O staff selected performance statistics, via the PMAS (report to include charts and graphs).

Test Outputs

The expected results of this test include:

An entry in the history log for each login attempt (valid or invalid) and a time stamp of when the action occurred.

An entry for the activation of the application with a corresponding start time and an entry for the termination of the application with a corresponding stop time.

An entry for the crossing of a threshold.

Science Algorithm performance data from the History Logs.

A database that maintains all of the desired information from the history log file.

Reports based on the results of the PMAS provided queries (to include charts and graphs).

Stored performance data within the PMAS.

A printout of the M&O staff-selected performance statistics.

Success Criteria

This test will be deemed successful when the history log is ingested to the database and the information in the history log corresponds to the information in the database.

Assumptions and Constraints

None.

4.3.10 Test Case 3.10: Management Data Access (TC013.004)

The purpose of the management data access test is to verify that a user can schedule the transfer and loading of the log files and an application can load the files into the management database. Verify that the database is accessible and that the integrity of the management data is maintained.

Test Inputs

Inputs to this test case include: management data, setting up a schedule to transfer and loading of the log files.

Test Steps

Access management data (verify that CSS services where used)

Transfer management data (verify that CSS Services where used)

Using an application access the management data

Repeat above step for selective data

Schedule the transfer and loading of log files

When the scheduled time is reached verify that the log files were transferred and loaded

Using an application verify that you can load the log files

Verify that throughout all of the above steps the data integrity was maintained

Test Output

Outputs to this test case include: Accessed and transferred management data, selectively accessed data and loaded log files.

Success Criteria

This test will be deemed successful when data has been transferred and loaded using both predetermined schedules and an application, log files were loaded and the integrity of the data was maintained.

Assumptions and Constraints

None.

4.3.11 Test Case 3.11: Performance Monitoring Thresholds (TC013.005)

The objective of this test case is to demonstrate the ability of the MSS Performance Management Application Service to provide a number of configurable thresholds for each performance metric, provide default values, allow for the modification of these values, and compare received values against these thresholds.

- Memory Threshold Fault

  This test verifies that the HP OpenView Network Node Manager detects and locates the computer with the memory threshold fault. Since most computers utilize a portion of their hard disk as virtual memory, the Management Information Base object created to monitor a computer's memory will contain threshold limits to determine when the memory capacity of a computer is in excessive use. (Test will be verified on all workstations/PCs at each site.)

- CPU Threshold Fault

  This test verifies that the HP OpenView Network Node Manager detects and locates the computer where the CPU threshold limit has been exceeded. The Management Information Base object created will contain threshold limits to determine when the CPU load of a computer is in excessive use. (Test will be verified on all workstations/PCs at each site.)

- Hard Disk Capacity Fault

  This test verifies that the HP OpenView Network Node Manager detects and locates the computer with a storage capacity fault. To simulate full capacity on a storage device, the tester will store large postscript or HDF files to the storage device. This test is limited to workstations and PCs with less than 1 GB of storage. (Test will be verified on all workstations/PCs at each site.)

Test Inputs

Inputs to this test case include HP OpenView Network Node Manager commands and ECS performance data.

A list of initial thresholds, including memory and CPU threshold limits.

Root map of the HP OpenView Network Node Manager window will be active in the tester's display.

Test Steps

Verify that the PMAS provides a configurable number of thresholds for each performance metric.

Verify that the EMC PMAS can create and send a list of suggested initial thresholds for each performance metric to the MSS site performance management application via CSS services and that the sites can receive it.

Examine the above list to ensure that a suggested initial threshold value exists for each performance metric.

Execute an application that exceeds the memory threshold limit determined within the Management Information Base object created to monitor the computer memory usage. (If threshold limit is too high to exceed, lower the threshold as needed. After test, threshold limit will be returned to original state.)

Execute an application that utilizes all/most of the CPU processing time.

Store as many large postscript and HDF files to the system storage device as possible without damaging any existing files on the system.

Ensure that the proper alarms/warning have been disseminated concerning any values exceeding their thresholds.

Reconfigure these threshold values to higher settings.

Reconfigure these threshold values to lower settings.

Test Outputs

Proper alarms/warning have been disseminated concerning any values exceeding their thresholds.

Internet symbol on Root map of the HP OpenView Network Node Manager window has turned YELLOW.

Traversing through the Internet submaps, following the YELLOW/marginal status symbols, the tester should be directed to the computer (indicated by color of RED) that contains the memory threshold fault, where the CPU threshold limit has been exceeded, or that contains the storage device fault (a message indicating storage capacity should also be displayed).

Success Criteria

This test will be deemed successful when all performance parameters exceeding their configured thresholds are flagged.

Assumptions and Constraints

None.

4.3.12 Test Case 3.12: Basic Monitoring (TC014.001)

The purpose of the Basic Monitoring test is to demonstrate the management framework's ability to create and display graphical representations of network topologies and to organize the given network topology into a hierarchy of maps.

- DAAC Hardware Confirmation

  This test verifies that the HP OpenView Network Node Manager properly detects and monitors all Ir1 hardware located at the Ir1 sites (GSFC, MSFC, LaRC, EDC and the EDF).

- DAAC Software Process Monitoring

  This test verifies that the MIB created within the HP OpenView Network Node Manager properly monitors all software processes at the local sites (GSFC, MSFC, LaRC, EDC and the EDF).

- Generation, Collection, Storing, and Displaying of Network Statistics

  This test verifies that the HP OpenView Network Node Manager is able to generate, collect, store, and display network statistics.

Test Inputs

Inputs to this test include the SNMP (the ECS standard protocol as specified in RFC 1157) managed objects.

Local site submap of HP OpenView Network Node Manager is active in the tester's display. Contact site liaisons and receive a list of all hardware components and software processes that are active, in-use, and should be monitored.

Request to display network statistics for GSFC, MSFC, LaRC, EDC & EDF systems.

Test Steps

Initialize HP OpenView and verify that a map depiction of the network topology is accurately displayed

Verify that the lower level topologies include hosts, routers, network interface cards, bridges, gateways, operating systems, peripherals, databases and their status.

Double click on the available icons to verify that lower level submaps exist

Determine the operational state of all network components, hosts and peripherals.

Force an operational state of a network component to change (repeat for hosts, applications, and peripherals in all of the following modes: on-line, off-line, and test).

Register and unregister a new network component.

Demonstrate the ability of the PMAS to receive notification of the change.

Demonstrate the capability of the PMAS to monitor the performance of network components (repeat for hosts, operating systems, peripherals, and databases).

Demonstrate the ability to generate reports on an interactive and scheduled basis to all of the following: console, disk file, and printer.

Demonstrate the capability of the performance management application service to monitor ECS component protocol stack performance parameters and Ethernet-like device performance parameters as defined in IETF RFC 1213 and IETF RFC 1623, respectively.

Repeat above steps locally at each site.

Test Outputs

Local site submap of HP OpenView Network Node Manager displays all hardware active at each site. Listing of all hardware and software processes active and in-use at each site.

Network statistics are displayed and stored for GSFC, MSFC, LaRC, EDC & EDF systems.

Success Criteria

Hardware and software processes displayed in local site's submap of HP OpenView Network Node Manager matches/confirms hardware and software processes that is obtained from the site liaison.

Assumptions and Constraints

None.

4.3.13 Test Case 3.13: OpenView (TC014.002)

The purpose of the OpenView test is to demonstrate the accuracy in which OpenView can monitor the changes that take place over the network and its ability to notify the Systems Administrator. The following series of tests verifies that OpenView will properly detect and locate hardware faults (computer, gateway/router, printer, peripheral, etc.) that are connected within the LAN/WAN network. Hardware faults may be caused by power loss or a network disconnect. The tests will also verify that OpenView will properly detect and locate software application initialization and termination, as well as perform protocol testing. All events should be recorded in a problem log.

Test Inputs

Inputs to this test case include initialization/termination of a software application that is being monitored by SNMP (Simple Network Management Protocol), hardware power loss and hardware connection/disconnection from the network.

Test Steps

Log onto a workstation, and initialize HP OpenView.

Verify that all objects/applications that are being monitored by SNMP are visible from the display.

Initialize an application being monitored by OpenView, and verify that the system recognizes the monitoring of the application.

Exit from the application, and verify that the system depicts the change.

Verify that OpenView can perform an IP protocol test.

Verify that OpenView can perform an TCP protocol test.

Verify that OpenView can perform an SNMP protocol test.

Verify that OpenView can perform an UDP protocol test.

Verify that OpenView can perform an ICMP protocol test.

Connect a hardware device (e.g. printer) to the network and verify that the system recognizes the new configuration.

Turn off the power to the hardware device (e.g. computer, gateway/router) and verify that the system recognizes the new configuration.

Turn the power to the hardware device back on and verify that the system recognizes the new configuration.

Disconnect the hardware device from the network and verify that the system recognizes the new configuration.

Change to the directory which contains the history log.

Examine the history log to determine whether all appropriate events have been documented.

Verify that with all of the network changes that occur an E-mail message was sent to the System Administrator.

Test Outputs

The expected results of this test include both visual and E-mail notification to the System Administrator. The history log file will be analyzed to determine that all network changes were logged properly.

Success Criteria

This test will be deemed successful when all network changes have been detected and logged and the Systems Administrator notified for each change as part of the dialog session.

Assumptions and Constraints

None.

4.3.14 Test Case 3.14: Fault Indication (TC014.003)

The objective of the Fault Indication test is to determine if a fault is categorized into proper severity levels.

- Multi-Process Termination Fault

  This test verifies that the HP OpenView Network Node Manager detects and locates the computer and processes that have been abruptly terminated. In this test many processes will be activated (GUIs, ftps, etc.), then some of the processes will be abruptly 'killed' simulating failures. Sample processes to kill include SNMP agents, DCE servers, ECS servers, etc.

Test Inputs

Inputs to this test case include setting the degree of a fault and modifying the time frame in which data will be gathered.

Root map of the HP OpenView Network Node Manager window will be active in the tester's display.

Test Steps

Initialize OpenView

Set the fault category for a particular fault

Re-configure the time in which the data will be gathered

Activate many processes on a host machine. (FTPs, Telnet, GUIs, etc.) 'Kill' a quarter of the processes from the machine.

Monitor the activity to verify that the system has accepted the changes

Display the information stored in the OpenView file

Test Outputs

The expected outputs include a display of the data gathered at the times indicated.

Internet symbol on Root map of the HP OpenView Network Node Manager has turned YELLOW.

Traversing through the Internet submaps, following the YELLOW/marginal status symbols, the tester should be directed to the computer and processes that were terminated.

Success Criteria

This test will be deemed successful when the fault categories we set have been implemented and the information is gathered for the predetermined times.

Assumptions and Constraints

None.

4.3.15 Test Case 3.15: MUI Services (TC014.004)

The purpose of the MUI (Management User Interface) Services test is to verify that the MUI Service provides all of the desired capabilities.

Test Inputs

Inputs to this test case include mouse and keyboard inputs, adding, deleting and modifying of symbols (shape, color and position), add, delete and modification of text string, M&O screen configuration changes, staff alert, vendor MIBs, managed objects and managed applications.

Test Steps

Initialize OpenView.

Launch an Xterm from the MUI to establish a dialog session with the M&O staff.

Demonstrate the MUI services compatibility with the ECS management framework.

Inspect the MUI vendor documentation to verify compliance with OSF/MOTIF.

Demonstrate using keyboard and mouse inputs the MUI service capability to respond.

Verify the capability of the M&O staff and an application to add/delete a symbol and modify the shape, color and position of the symbol.

Demonstrate the capability of the M&O staff and an application to add/delete and modify text strings.

Acting as an M&O user, verify that you can make screen configuration changes and save these changes.

Verify that as both M&O and an application you can load and unload vendor MIBs.

Verify that an M&O user has the capability to register and unregister managed objects and management applications.

Verify that an application can register and unregister managed objects or processes.

Verify capability to configure/customize event notifications.

Verify that the MUI provides on-line help windows for the applications.

Verify that the PMAS is capable of graphically displaying the operational state of managed objects in graphical form via the MUI service.

Verify that the PMAS is capable of displaying M&O staff-selected performance statistics in graphical and tabular form through the MUI service.

Print the performance statistics.

Test Outputs

Outputs to this test case include: responses from the MUI to the keyboard and mouse inputs, additions, deletions and modifications of symbols, add, delete and modification of text strings, new screens upon configuration changes, vendor MIBs, managed objects and management applications and on-line help windows. Graphical displays of the operational states of managed objects and graphical and tabular displays of M&O staff-selected performance statistics. A printout of the performance statistics.

Success Criteria

This test will be deemed successful when all of the represented test steps have been executed and the appropriate results were obtained.

Assumptions and Constraints

None.

4.3.16 Test Case 3.16: Performance Management (TC014.005)

The purpose of the Performance Management test is to verify that all of the documented capabilities and requirements are tested and verified. The following data will be retrieved for network component interfaces: operational status, type, speed, octets in/out, packets in/out, discards in/out, and errors in/out. The following data will be retrieved for network hosts: total CPU utilization, memory utilization, physical disk in/out, disk storage size, disk storage used, number of active processes, length of run queue, network in/out (packets), and network errors.

Test Inputs

Inputs to this test case include: managed objects, performance metrics and requested performance data.

Test Steps

Demonstrate the ability of the PMAS to manage an object.

Verify that the PMAS is capable of receiving managed object definitions for each managed object.

Gather data for various performance metrics from each individual managed object, verify that you are able to specify which metrics you would like to gather.

Demonstrate the ability of the PMAS to request data from the individual managed objects on a configurable interval and on demand.

Verify that the PMAS can receive requested performance data from the ECS components.

Demonstrate the ability of the PMAS to retrieve data for all hosts, peripherals and network component interfaces.

Demonstrate the ability to generate reports on an interactive and scheduled basis to all of the following: console, disk file, and printer.

Test Outputs

Reports of performance metrics, requested data from individual objects, requested performance data.

Success Criteria

This test will be deemed successful when the PMAS meets all of the functionality outlined above.

Assumptions and Constraints

None.

4.3.17 Test Case 3.17: Monitor/Control and Management Agent (TC014.006)

The purpose of the Monitor/Control and Management Agent test is to verify the functionality provided by the Management Agent Service and the Monitor/Control Service, as well as their ability to interact with each other.

Test Inputs

Inputs to this test case include: ECS management set messages, traps/events, performance and fault data, data from ECS managed objects, and a proxy agent for ECS network devices.

Test Steps

Demonstrate the communications via the ECS management protocol between the Monitor/Control Service and the Management Agent Service in test or operational mode.

Verify that the Monitor/Control Service can send ECS management set messages to configure and control the processing performed by the ECS management agent and receive ECS management traps/events.

Demonstrate the functionality associated with the Monitor/Control Service to perform statistical analysis on the performance and fault data collected from the Management Agent Service.

Verify that the MSS Management Agent Service retrieves data from ECS managed objects in test or operational mode.

Verify that the Management Agent Service can respond to requests for managed object MIB attributes.

Verify the capability to browse MIB values.

Verify that the Management Agent Service can receive ECS management traps/events and set messages from the Monitor/Control Service.

Demonstrate the ability of a Management Agent Service provided ECS management agent to be configurable to include: community to respond to and set attributes, agent location and contact person, traps to send and events to log and log file name.

Verify that an ECS management agent is provided for network devices.

Verify that for all network devices and applications, which can not be managed via SNMP, the Management Agent Service provides proxy agents.

Test Outputs

Outputs to this test case include, but are not limited to: communications between the Monitor/Control Service and Management Agent service, management traps/events, statistical analysis, retrieved data from the ECS managed objects, a configured management agent and proxy agents.

Success Criteria

This test will be deemed successful when communications between the Management Agent Service and Monitor/Control Service and all of the documented functionality has been verified.

Assumptions and Constraints

None

4.3.18 Test Case 3.18: Problem Tracking Test (TS002.011)

This test case demonstrates the ability to track problems that are identified in the configured Ir1 environment.

Test Inputs

Inputs to this test are simulated problems found in the Ir1 environment.

Test Steps

See Section 5

Test Outputs

Outputs to this test are Problem Tracking reports.

Success Criteria

This test is deemed successful if all simulated problems found in the Ir1 environment are tracked and problem reports are generated.

Assumptions and Constraints

None

4.3.19 Test Case 3.19: Remote NCR (BC016.003)

This test case demonstrates the capability of the DAAC M&O Representative to log a NCR (Nonconformance Report).

Test Inputs

Inputs to this test case include the submittal of an NCR by a user located at one of the DAACs.

Test Steps

Have the system administrator create a test account on one of the DAACs.

Log into the DAAC as that user.

Using the Discrepancy Tracking System document a discrepancy.

Verify that the NCR is logged.

Verify that appropriate E-mail notifications are sent out through DDTS.

Track the progress of the NCR through verification.

Test Outputs

The expected results of this test include the ability of a DAAC user to record an NCR into the Discrepancy Tracking System. A printout documenting the discrepancy should be generated.

Success Criteria

This test will be deemed successful when a user located at the DAAC can log a discrepancy report and track the success of solving the discrepancy.

Assumptions and Constraints

None.

4.3.20 Test Case 3.20: Hardware Monitoring Process Terminated (T04-01.01.06)

This test verifies that the System Management Framework tool will detect and locate a hardware "fault" when the monitoring process of the specific piece of hardware has been terminated. Testing includes monitoring the System Management Framework tool while the monitoring process is terminated, therefore, a System Management Framework window will be displayed on the workstation/PC where the tester is located. (Test will be verified at each site.)

Test Inputs

Root map of System Management Framework window will be active in the tester's display. Tester will locate and terminate/"kill" the process that monitors the hardware.

Test Steps

See Section 5

Test Output

Internet symbol on Root map of System Management Framework has turned YELLOW.

Success Criteria

Traversing through the Internet submaps, following the YELLOW/marginal status symbols, the tester should be directed to the "faulty" piece of hardware (indicated by color RED).

Assumptions and Constraints

None.

4.3.21 Test Case 3.21: Gateway/Router Monitoring Process Terminated (T04-01.01.09)

This test verifies that the System Management Framework tool detects and locates a possible fault with a gateway/router. Testing includes displaying the System Management Framework tool, while the tester terminates the process that monitors the gateway/router activities. (Test will be verified at each site.)

Test Inputs

Root map of System Management Framework window will be active in the tester's display. Tester will locate and "kill" the process that monitors the gateway/router.

Test Steps

See Section 5

Test Output

Internet symbol on Root map of System Management Framework has turned RED.

Success Criteria

Traversing through the Internet submaps, following the RED/critical status symbols, the tester should be directed to the faulty gateway/router (color should be RED).

Assumptions and Constraints

None.

4.3.22 Test Case 3.22: Software Application Monitoring Process Termination (T04-01.01.12)

This test verifies that the System Management Framework tool detects and locates the terminated software application when the monitoring process of the application has been terminated. Testing includes monitoring the System Management Framework tool while the monitoring process is terminated, therefore, a management window will be displayed on the workstation/PC where the tester is located. (Test will be verified at each site.)

Test Inputs

Root map of System Management Framework window will be active in the tester's display. Tester will locate and terminate/"kill" the process that monitors the software application.

Test Steps

See Section 5

Test Output

Internet symbol on Root map of System Management Framework has turned YELLOW.

Success Criteria

Traversing through the Internet submaps, following the YELLOW/marginal status symbols, the tester should be directed to the computer where the application was running (computer will be RED in color) and the application itself.

Assumptions and Constraints

None.

4.3.23 Test Case 3.23: Computer Monitoring Process Terminated (T04-01.01.18)

This test verifies that the System Management Framework tool will detect and locate a computer "fault" when the monitoring process of the computer has been terminated. Testing includes monitoring the System Management Framework tool while the monitoring process is terminated, therefore, a System Management Framework window will be displayed on the workstation/PC where the tester is located. (Test will be verified at each site.)

Test Inputs

Root map of System Management Framework window will be active in the tester's display. Tester will locate and terminate/"kill" the process that monitors the computer.

Test Steps

See Section 5

Test Output

Internet symbol on Root map of System Management Framework has turned YELLOW.

Success Criteria

Traversing through the Internet submaps, following the YELLOW/marginal status symbols, the tester should be directed to the "faulty" computer (indicated by color or RED).

Assumptions and Constraints

None.

4.3.24 Test Case 3.24: Operating System Monitoring Process Terminated (T04-01.01.20)

This test verifies that the System Management Framework tool detects and locates the computer with the "failed" operating system. To simulate a failed operating system the monitoring process of the system will be terminated/"kill"ed, no processing will be harmed. Testing includes monitoring the System Management Framework tool while the monitoring process is terminated. (Test will be verified on all computers at each site.)

Test Inputs

Root map of System Management Framework window will be active in the tester's display. Opening an xterm window from a host machine, locate and "kill" the process that monitors the operating system.

Test Steps

See Section 5

Test Output

Internet symbol on Root map of System Management Framework has turned YELLOW.

Success Criteria

Traversing through the Internet submaps, following the YELLOW/marginal status symbols, the tester should be directed to the computer and actual operating system fault that occurred (Computer will be RED in color).

Assumptions and Constraints

None.

4.3.25 Test Case 3.25: Local Site Management/Security Policy and Procedures (T04-01.02.03)

This test verifies that local site security policy & procedures including password management, operational security, data classification, compromise mitigation and access/privileges, systems hardware and software maintenance, are current and updated at each site supported by this release.

Test Inputs

Security management policies and procedures.

Test Steps

See Section 5

Test Output

Security management policies and procedures from GSFC, MSFC, EDC and LaRC.

Success Criteria

Security sections within all documents are current and identical to that copy held at the EDF.

Assumptions and Constraints

Ir1 sites that do not have established policies and procedures, will be able to generate and distribute them through the OA tools..

4.3.26 Test Case 3.26: Active DAAC ECS Administrator Account (T04-01.05.01)

An active Administrator account exists to provide a maintenance and operational interface to the DAACs to allow resource usage and management.

Test Inputs

Account with special accesses assigned to System Administrator.

Test Steps

See Section 5

Test Output

Account exists with necessary privileges.

Success Criteria

Interface account for system exists.

Assumptions and Constraints

None.

4.3.27 Test Case 3.27: ECS Software Backup Maintained (T04-01.05.02)

A minimum of one backup save set is maintained in a separate physical location of the ECS software.

Test Inputs

Management policies and procedures guidelines from the EDF.

Test Steps

See Section 5

Test Output

Management policies and procedures manual from GSFC, MSFC, EDC and LaRC.

Success Criteria

Section within document are current and identical to that copy held at the EDF.

Assumptions and Constraints

None.

4.4 System Administration Build Test

The System Administration Build Test represents an aggregation of the DCE Infrastructure, Messaging and File Transfer, and System Management threads. The functions to be tested include general DCE functionality, network management, fault management, security management, and Ir1 internetworking capabilities.

Special resources required for this build test include:

- Cell Directory Service Command Program (cdscp)
- XRunner
- LoadRunner
- Privileges to execute DTS management functions
- Sample science data files
- HP OpenView
- Network analyzer

4.4.1 Test Case 4.1: General DCE (BC008.001)

The purpose of the General DCE test is to demonstrate, that upon integration of the custom code developed for the DCE Infrastructure, Messaging and File Transfer, and System Management Threads, all of the expected capabilities will remain intact.

Test Inputs

Inputs to this test case include a valid ID and valid password, various combinations of valid/invalid ID and valid/invalid password, valid admin ID and password, valid add, change and delete registry commands, ability to access and modify directories, time checks using DTS.

Test Steps

Run an XRunner script to execute a number of login attempts using a combination of valid/invalid IDs and passwords.

Upon successful login, call another XRunner script to change the users password, logout and login with the new password

Create a cron job that will logon to each of the available Ir1 workstations and record the distributed time in a flat file

Start the cron job running in the background

Logon as the DCE Administrator

Add, change, and delete commands to/from the security registry

Set up a cron job to retrieve the time from each of the workstations in the Ir1 operational cell and store them in a file

Inspect the file to insure that all of the times are in sync

Set up a workstation to run as the server

Set up all of the workstations to be clients (including the server workstation)

Initialize the server

Initialize communications between client and server.

View the flat file containing the times from the various workstation to ensure distributed time

Test Outputs

Screen outputs showing the success or failure of the logon/logoff attempts. Response times of each logon and logoff event. Network monitor output showing the data transmitted between client and server. Event log data. Flat file showing the times recorded during execution of the cron job. Screen outputs showing successful bindings between clients and server.

Success Criteria

The test will be deemed successful when all Ir1 DCE functionality have been demonstrated and verified.

Assumptions and Constraints

None.

4.4.2 Test Case 4.2: Network Management Test (BC002.004)

This test case demonstrates that the internetworking devices do event notification of relevant networking events as follows:

Detect the following types of faults, errors, and events:

- communication software version mismatch errors
- communication software configuration errors
- communication hardware errors
  - host not reachable
  - router not reachable
  - errors and failures of communication links
- errors in supported communication protocols
- degradation of performance due to exceeded thresholds
- peripherals

- databases
- applications
  - process missing (application or COTS product)
  - process in a loop
  - process failed

Generate the following types of notifications:

- change in displayed icon color
- pop-up window notification
- logging following fault information to disk log file:
  - fault type
  - date and time of fault
  - ID of notification source (e.g. IP address, process name, etc.)
  - fault data received with the notification
  - operator-defined descriptive text
- audible alert

Test Inputs

Network manager commands to activate network device agents.

Test Steps

Run network benchmarks with network event logging turned on.

Use network manager to requests various types of event data from network devices.

Introduce anomalies into the system, for example: causing the failure or interruption of various network devices, causing thresholds to be exceeded.

Verify that logs are populated with appropriate information and that E-Mail messages are sent to system administration.

Conduct basic fault isolation analysis through HPOV diagnostics and testing of:

- connectivity between pairs of ECS hosts and ECS routers
- ability to reach hosts and routers
- availability of network services at hosts

Repeat above steps for a software configurable item.

Test Outputs

Logs and alarms showing recorded event data.

Success Criteria

Log and alarm data consistent with the above Steps. No spurious alarm or log data.

Assumptions and Constraints

This functionality will be further verified in other build/thread tests since internetworking will implicitly be a part of most other tests.

4.4.3 Test Case 4.3: Fault Management (T04-01.02.01)

This test verifies that the fault management policies and procedures located at GSFC, MSFC, EDC and LaRC are current and consistent with those located at the EDF.

Test Input

Current fault management policies and procedures from the EDF. Fault management policies and procedures from GSFC, MSFC, EDC and LaRC.

Test Steps

See Section 5

Test Output

Fault management policies and procedures from GSFC, MSFC, EDC and LaRC.

Success Criteria

Fault sections within all documents are current and identical to that copy held at the EDF.

Assumptions and Constraints

None.

4.4.4 Test Case 4.4: Security Management (T04-01.02.02)

This test verifies that the security management policies and procedures located at GSFC, MSFC, EDC and LaRC are current and consistent with those located at the EDF.

Test Inputs

Security management policies and procedures from the EDF.

Test Steps

See Section 5

Test Output

Security management policies and procedures from GSFC, MSFC, EDC and LaRC.

Success Criteria

Security sections within all documents are current and identical to that copy held at the EDF.

Assumptions and Constraints

None.

4.4.5 Test Case 4.5: Access to GSFC (T01-02.02.02)

This test verifies that all host machines connected to the LAN at GSFC are accessible through the network from the EDF.

Test Inputs

Listing of all machines connected to GSFC LAN. "ping" all host machines on listing.

Test Steps

Log onto V0 host machine.

Ping statistics from V0 host machine.

Test Outputs

Ping statistics displayed for each machine.

Success Criteria

All host machines listed from GSFC should return ping messages that are connected to GSFC LAN. History log will record all activities.

Assumptions and Constraints

None.

4.4.6 Test Case 4.6: Access to LaRC (T01-02.02.03)

This test verifies that all host machines connected to the LAN at LaRC are accessible through the network from the EDF.

Test Inputs

Listing of all machines connected to LaRC LAN. "ping" all host machines on listing.

Test Steps

Log onto V0 host machine.

Ping statistics from V0 host machine.

Test Outputs

Ping statistics displayed for each machine.

Success Criteria

All host machines listed from LaRC should return ping messages that are connected to LaRC LAN. History log will record all activities.

Assumptions and Constraints

None.

4.4.7 Test Case 4.7: Access to MSFC (T01-02.02.04)

This test verifies that all host machines connected to the LAN at MSFC are accessible through the network from the EDF.

Test Inputs

Listing of all machines connected to MSFC LAN. "ping" all host machines on listing.

Test Steps

Log onto V0 host machine.

Ping statistics from V0 host machine.

Test Outputs

Ping statistics displayed for each machine.

Success Criteria

All host machines listed from MSFC should return ping messages that are connected to MSFC LAN. History log will record all activities.

Assumptions and Constraints

None.

4.4.8 Test Case 4.8: Internetworking Test (BC002.002)

This test case tests internetworking services for TCP/IP and UDP/IP over Ethernet and FDDI.

Test Inputs

Benchmark internetworking test data.

Test Steps

Run internetworking benchmark test suite for TCP/IP and UDP/IP protocols.

Test Outputs

Comparisons of test data sent and received, network analyzer outputs, log data indication performance and fault data, and network manager alarms.

Success Criteria

Data successfully transported through the network, performance data consistent with prior and standard benchmark results, no anomalous performance or fault log or alarm events.

Assumptions and Constraints

This functionality will be further verified in other build/thread tests since internetworking will implicitly be a part of most other tests.

4.5 AI&T Tools Thread Test

The purpose of this thread is to verify the functionality of the Ir1 AI&T tools. To support AI&T in the evaluation of science software, these tools should provide for: documentation viewing, standards checking, code execution using make files, results comparison, configuration management, report generation, and display of product metadata and on-line documentation.

Special resources required for this thread test include:

Hardware – workstation, printer, local disk storage

Software – Documentation Viewing tools, Standards Checkers and Report Generation tool, File Comparison utility, Product Metadata Display tool, Profiling tools

Data – PostScript, ASCII, HTML and RTF files, FORTRAN77 code, Science Software code, Ada code, product metadata, Science Software Delivery software (shell scripts, executables)

Test Tools - Xrunner, Loadrunner

4.5.1 Test Case 5.1: Viewing Science Software Documentation Test (TS002.002)

This test case demonstrates the capability to display and print documents included in a science software delivery in PostScript, ASCII, Hypertext Markup Language (HTML), Microsoft Word, Word Perfect and Adobe Acrobat Portable Document Format (PDF).

Test Inputs

Documents (in ASCII, HTML, Microsoft Word, Word Perfect and PDF) of science software in valid and invalid formats.

Test Steps

See Section 5

Test Outputs

Outputs to this test include legible displays and printouts of documents in valid formats.

Success Criteria

Documents in valid formats should be displayable and printable.

Assumptions and Constraints

None.

4.5.2 Test Case 5.2: Code Standard Checking Test - FORTRAN77 Code (TS002.003)

This test case demonstrates the capability to:

(1) verify that all science software code contain headers and follow the coding standards established in the "Data Production Software and SCF Standards and Guidelines" document.

(2) verify the capability to generate report files describing the result of standards checking

Test Inputs

Inputs to this test include compliant and non-compliant FORTRAN77 code and scripts from the science software delivery.

Test Steps

See Section 5

Test Outputs

Outputs to this test include:

- FORTRAN77 code contain headers and follow the coding standards established in the "Data Production Software and SCF Standards and Guidelines" document
- Report files describing the result of compliance and standards checking

Success Criteria

- All FORTRAN77 code contain headers and follow the coding standards established in the "Data Production Software and SCF Standards and Guidelines" document.
- Generates report files describing the result of standards checking

Assumptions and Constraints

None.

4.5.3 Test Case 5.3: Code Standard Checking Test - FORTRAN90 Code (TS002.004)

This test case demonstrates the capability to:

(1) verify that all science software code contain headers and follow the coding standards established in the "Data Production Software and SCF Standards and Guidelines" document.

(2) verify that the capability to generate report files of standards checking.

(3) verify that Science software code is POSIX compliant

Test Inputs

Inputs to this test include compliant and non-compliant FORTRAN90 code and scripts from the science software delivery.

Test Steps

See Section 5

Test Outputs

Outputs to this test include:

- FORTRAN90 code contain headers and follow the coding standards established in the "Data Production Software and SCF Standards and Guidelines" document
- Report files describing the result of compliance and standards checking

Success Criteria

- All FORTRAN90 code contain headers and follow the coding standards established in the "Data Production Software and SCF Standards and Guidelines" document.
- All FORTRAN90 code are POSIX compliant.
- Generates report files describing the result of standards checking

Assumptions and Constraints

None.

4.5.4 Test Case 5.4: Code Standard Checking Test - "C" Code (TS002.005)

This test case demonstrates the capability to:

(1) verify that software and scripts included in a science software delivery and written in "C" code, are in compliance with ANSI, standard specification.

(2) verify the capability to generate report files of standards checking.

Test Inputs

Compliant and non-compliant "C" code and scripts from the science software delivery.

Test Steps

See Section 5

Test Outputs

Outputs to this test include:

- Report files describing the result of compliance and standards checking
- Status messages

Success Criteria

For each science software delivery containing "C" code, a report shall be generated that logs the adherence/non-adherence to POSIX and standard specification.

Assumptions and Constraints

None.

4.5.5 Test Case 5.5: Code Standard Checking Test - Ada Code (TS002.006)

This test case demonstrates the capability to:

(1) verify that Ada code and scripts included in a science software delivery are in compliance with POSIX and MIL-STD-1815-A standard specifications.

(2) verify that all Ada code contain headers as specified in 423-16-01, and the "Data Production Software and SCF Standards and Guidelines.

(3) verify the capability to generate report files of standards checking.

Test Inputs

Inputs to this test are compliant and non-compliant Ada code and scripts from the science software delivery.

Test Steps

See Section 5

Test Outputs

- Report files describing the result of compliance and standards checking
- Status messages

Success Criteria

For each science software delivery containing Ada code, a report shall be generated that logs the adherence/non-adherence to POSIX and MIL-STD-1815-A standard specifications and "Data Production Software and SCF Standards and Guidelines" document (423-16-01).

Assumptions and Constraints

None.

4.5.6 Test Case 5.6: Code Standard Checking Test - PGS Toolkit Usage Requirements (TS002.007)

This test case demonstrates the capability to:

(1) verify that Science Software source code and Science Software scripts follow the following PGS toolkit usage requirements (from 194-809-SD4-001, PGS toolkit Users Guide for the ECS Project):

- Source code does not make any prohibited POSIX function calls.
- The Status Message Text Files have the correct format.
- None of an operator-specified set of Toolkit functions is called more than once per PGE.
- Of an operator-specified set of Toolkit functions, none is called by a PGE prior to another operator-specified set of Toolkit functions.

(2) verify that the standards checking capabilities include the following:

- Flagging whenever a bit operation is used on signed numbers (C only).
- Flagging argument list mismatches (type and number of arguments).

(3) verify that the first line of each science software script specifies either the "C", Bourne, Korn, Perl, or POSIX shell

(4) verify that all science software code contain headers and follow the coding standards established in the "Data Production Software and SCF Standards and Guidelines" document

(5) verify the capability to generate report files of standards checking

Test Inputs

Inputs to this test include standard and non-standard science software and scripts deliveries.

Test Steps

See Section 5

Test Outputs

Outputs to this test include status messages, logs and analyzed science software code.

Success Criteria

Science software code should pass all standards checking without errors.

Science Software source code and Science Software scripts should follow the PGS toolkit usage requirements without errors.

Assumptions and Constraints

None.

4.5.7 Test Case 5.7: File Comparison Test (TS002.010)

This test case demonstrates the ability to read ASCII, binary, HDF or user defined custom data format and find all differences existing between any two files within a specified absolute or relative threshold and generate reports of the file comparison results.

Test Inputs

The inputs to this test include ASCII, binary, HDF and user defined custom data formats.

Test Steps

See Section 5

Test Outputs

The outputs to this test include status logs, reports and files of data differences within a specified tolerance.

Success Criteria

Given two data files and a specified tolerance, existing differences shall be displayed in status, log, file or report.

Assumptions and Constraints

None.

4.5.8 Test Case 5.8: Status Tracking and Report Generation Test (TS002.012)

This test case demonstrates the ability to track the status of the science software integration process, and to demonstrate the ability of authorized users to examine the produced logs and reports at any time from receipt through the end of acceptance. The status tracking processing shall entail the usage of plotting tools, spreadsheets, drawing and word processors to produce logs, reports and status information (according to specified templates) throughout the integration phase. In addition, demonstrate the ability to report on the status of I &T related discrepancy reports.

Test Inputs

Inputs to this test should include status tracking logs, spreadsheets and I & T related discrepancy.

Test Steps

See Section 5

Test Outputs

- Outputs to this test should include new and updated status tracking logs and spreadsheets.
- Report on the status of I & T - related discrepancy reports

Success Criteria

- Successful tracking of the status of science software integration process
- Successful production of logs and reports through the end of acceptance
- Successful generation or report on the status of I & T - related discrepancy/reports

Assumptions and Constraints

None.

4.5.9 Test Case 5.9: Product Metadata Test (TS002.015)

This test case verifies the capability to view metadata associated with a data file and write it to a report file.

Test Inputs

Inputs to this test include data file metadata.

Test Steps

See Section 5

Test Outputs

The output to this test includes legible metadata file displays and report files.

Success Criteria

All data file metadata shall be able to produce a legible report file via the use of the PGS toolkit.

Assumptions and Constraints

None.

4.5.10 Test Case 5.10: Profiling Test (TS002.016)

This test demonstrates the ability of the performance analysis and resource management tools to measure, record and report the following performance statistics of a process: CPU time, CPU time of each procedure, memory usage, disk space usage, number of I/O accesses to each of its input or output data files, wall clock time of a process, wall clock time of a procedure, and number of page faults. These statistics can be printed or saved as soft copy.

Test Inputs

AI&T performance metadata, AIT resource metadata. A series of tests are performed using data which produces varying results (CPU time CPU time of each procedure, memory usage, disk space usage, and the number of I/O accesses to each input or output data files).

Test Steps

See Section 5

Test Outputs

Report which chronicles the performance statistics of a designated process which can be displayed to console, printed to hard copy or saved as soft copy.

Success Criteria

For all performance/resource metadata regarding a specific process submitted, a report will be produced of performance statistics. Statistics will be verified for accuracy.

Assumptions and Constraints

None.

4.5.11 Test Case 5.11: Process (Check-in) Algorithm to Configuration Management (T03-01.05.01)

This test verifies the ability to bring the algorithms received from the science community under CM control. It also verifies the ability to access and display vital information for the CMed algorithms.

Test Inputs

Science Algorithms and all associated data to include source code, author, benchmark test procedures/data/results, and compiler version/identification. Designated CM repository and CM Tool. Network environment to support file transfer or Tape I/O capability.

Test Steps

See Section 5

Test Outputs

Appropriate status/error messages, version information and log entries generated as a result of loading science algorithm into CM using the CM Tool.

Success Criteria

Version 1 of the Science Algorithms and all its associated data should be resident on the designated CM repository. CMed algorithms should display version information (Audit trail) and be accessible by all valid users.

Assumptions and Constraints

None.

4.5.12 Test Case 5.12: Check-Out Algorithm from Configuration Management To User (T03-01.05.02)

This test verifies the ability to check an algorithm under CM Control out to the science community or other users.

Test Inputs

Designated CM repository and CM Tool. Algorithms resident under CM Control. Network environment and valid users that can "check-out" an algorithm from CM.

Test Steps

See Section 5

Test Outputs

Appropriate status/error messages and log entries generated as a result of checking science algorithm out from CM using the CM Tool.

Success Criteria

Version logged out will reflect "checked out" to "xxx user".

Assumptions and Constraints

None.

4.5.13 Test Case 5.13: Check-Out Single Algorithm from Configuration Management To Multiple Users (T03-01.05.03)

This test verifies the ability to check a single algorithm under CM Control out to multiple users of the science community.

Test Inputs

Designated CM repository and CM Tool. Algorithms resident under CM Control. Network environment and valid users that can "check-out" an algorithm from CM.

Test Steps

See Section 5

Test Outputs

Appropriate status/error messages and log entries generated as a result of checking science algorithm out from CM using the CM Tool.

Success Criteria

Version logged out will reflect "checked out" to "xxx user". Multiple users should be able to check out the same version of an algorithm while only the first user to "check-out" the algorithm will have the "reserved' copy.

Assumptions and Constraints

None.

4.5.14 Test Case 5.14: Process (Check-in) Modified Algorithms to Configuration Management (T03-01.05.04)

This test verifies the ability to receive algorithms modified by the science community and check them into Configuration Management as an updated version using the designated CM Tool. In addition, it verifies the ability to provide an audit trail for the modifications.

Test Inputs

Original version for the modified Science Algorithm must already be resident in CM. Modified science algorithms and any updates to its supporting data (test files/documentation, etc.). CM repository and CM Tool. Network environment to support file transfer or tape I/O capability.

Test Steps

See Section 5

Test Outputs

Appropriate status/error messages and log entries generated as a result of loading modified science algorithm into CM using the CM Tool. Updated Version of science algorithm with audit trail information and any associated data resident on the designated CM repository. Original algorithm "checked-in" should not have changed.

Success Criteria

Original Version 1 of the Science Algorithm plus updated version (Version 1 plus DIFF to make up new version of the algorithm) resident under CM and accessible by all valid users. An audit trail information for updated version should be generated.

Assumptions and Constraints

None.

4.5.15 Test Case 5.15: Check-in of Single Algorithm with Multiple Check-Outs (T03-01.05.05)

This test verifies the ability to maintain CM control of system algorithms when one algorithm is checked out by more than a single user simultaneously.

Test Inputs

Original version of the Science Algorithm must already be resident in CM. CM repository and CM Tool. Multiple valid users to simultaneously "check-out" the same Science Algorithm. Network environment to support file transfer or TAPE I/O capability.

Test Steps

See Section 5

Test Outputs

Appropriate status/error messages and log entries generated as a result of multiple users checking the same algorithm out from CM utilizing the CM Tool. Appropriate status/error messages and log entries generated as a result of multiple users checking the same algorithm back into CM utilizing the CM Tool.

Success Criteria

Algorithm correctly reflects "checked-out" when an authorized user has obtained proper access to the algorithm through the CM Tool. First user to obtain copy of the algorithm will have the "reserved" copy. Any user that is NOT the first user to "check-out" the algorithm from CM should NOT be able to check the algorithm back into CM while the reserved copy remains "checked-out".

Assumptions and Constraints

None.

4.5.16 Test Case 5.16: Enhanced Algorithms Under Configuration Management at EDF (T03-01.05.06)

This test demonstrates that the SMC provides overall management of CMed enhanced Algorithms by retaining a master copy at the EDF.

Test Inputs

Original version of the science algorithm must be present in CM repository. Request is made to CM for enhanced version of the algorithm.

Test Steps

See Section 5

Test Outputs

Appropriate status/error message and log entries occurs when a request is made and a file is ready to be transferred to the requester. File is transferred by FTP from EDF to requester. An acceptance code is received from requester verifying the status of the delivery.

Success Criteria

SMC sends the enhanced version of the algorithm and maintains original version and the enhanced version in the CM library at the EDF.

Assumptions and Constraints

None.

4.5.17 Test Case 5.17: Check-In GFE, COTS, and/or Public Domain Database Files To CM (T03-01.06.01)

This test verifies the ability to check GFE, COTS, and/or Public Domain Databases into CM.

Test Inputs

Designated CM repository and CM Tool. GFE, COTS, or Public Domain database files to be "checked-in" to CM. Network environment to support file transfers or tape I/O capability.

Test Steps

See Section 5

Test Outputs

Appropriate status/error messages and log entries generated as a result of checking the database file into CM using the CM Tool.

Success Criteria

Version 1 of the GFE, COTS, and/or Public Domain database should be resident on the designated CM repository and be accessible by all valid users.

Assumptions and Constraints

None.

4.5.18 Test Case 5.18: Data Comparison Tool for Identical Algorithms (T03-01.08.01)

This test is to determine if a comparison tool adequately compares two identical algorithms. An algorithm is run in the SCF environment and deemed accurate. This algorithm is run in the DAAC environment. The results, which can be in either ASCII, binary, HDF, or user defined custom data formats, are compared using the algorithm comparison tool.

Test Inputs

An algorithm (and algorithm output) successfully run in the SCF environment. Access to data comparison tools. This algorithm is run in DAAC environments. The results from running the algorithm are used to test the comparison tool.

Test Steps

See Section 5

Test Outputs

Status messages and error reporting from the comparison tool.

Success Criteria

The tool should indicate no differences between algorithm results.

Assumptions and Constraints

None.

4.5.19 Test Case 5.19: Data Comparison Tool for Differing Algorithms (T03-01.08.02)

This test determines if a comparison tool adequately detects inconsistencies in output from two algorithms with known differences. An algorithm is run in the SCF and deemed valid. These same algorithm is modified and run again in the SCF. A comparison tool is used to compare the algorithm and the algorithm after modification. All outputs from the algorithms and modifications made to the algorithm are recorded. The SCF algorithm prior to modification and the SCF algorithm after modification is then run in a DAAC environment. Again the comparison tool is used to compare the algorithm results. Finally the output from the comparison tool used in the SCF is compared to the output from the comparison tool used in the DAAC.

Test Inputs

An algorithm and modified algorithm (and algorithm output) run in the SCF environment. Access to data comparison tools. This algorithms are run in DAAC environments. The results from running the algorithms are used to test the comparison tool.

Test Steps

See Section 5

Test Outputs

Algorithm output. Status and error reporting from the comparison tool.

Success Criteria

The tool should indicate differences based upon modifications made to the algorithm. The results from the comparison tool used at the SCF should be the same as the results obtained from the comparison tool used at the DAAC. SCF scientists will work with DAAC personnel to determine acceptable comparison tool output if differences are found.

Assumptions and Constraints

None.

4.5.20 Test Case 5.20: Data Comparison Tools for all ECS Defined Platforms (T03-01.08.03)

This test verifies that a data comparison tool will capture all differences between an algorithm run on different platforms. The same algorithm is run on two or more different platforms. Using a comparison tool the algorithm are compared. Differences are analyzed to determine if the results given by the comparison tool are complete and accurate.

Test Inputs

A valid algorithm is run on different platforms in the SCF. A comparison tool is used to compare results of the same algorithm run on different platform in the SCF. The results are recorded. The same algorithm is run on different platforms in the DAAC. Again a comparison tool is used to compare the results of the algorithm run on different DAAC platforms. The DAAC results are recorded.

Test Steps

See Section 5

Test Outputs

The results from the comparison tool run at the SCF are compared to the results recorded at the DAAC.

Success Criteria

There should little to no difference between SCF comparison tool output and DAAC comparison tool output. SCF scientists will work with DAAC personnel to determine acceptable comparison tool output if differences are found.

Assumptions and Constraints

None.

4.6 DAAC Toolkit Thread Test

This thread demonstrates that the DAAC version of the Product Generation System (PGS) toolkits are successfully used for the science software (algorithm) integration process through swapping of the Science Computing Facility (SCF) version of the PGS toolkit, for the DAAC version of the PGS toolkit for each science software delivery. The process by which the science software and PGS toolkit integration is verified at AI&T is a three stage process. The SCF version of the PGS toolkit tools and the science software are independently compiled, linked, then tested for functionality in the SCF environment. Upon delivery of the new science software to the DAAC, the SCF version of the PGS toolkit is again compiled, linked, and tested in the DAAC environment with the respective science software delivery. Providing test results are satisfactory, the DAAC version of the PGS toolkit is then compiled, linked, and tested with the same science software at the DAAC, thus assuring a safe and functional environment for the science software delivery. In order to ensure a high degree of portability across a wide variety of platforms the toolkit conforms to the Portable Operating System Interface for Computer Environments (POSIX) standards, enabling science software porting from the SCF to a DAAC environment (and through hardware changes as ECS matures).

Special resources required for this thread test include:

Hardware – Representative DAAC and SCF hardware and operating systems as outlined in Appendix B.

Software – DAAC and SCF versions of the PGS toolkit functions. They are as follows:

Status Message (SMF) tools

Process Control (PC) tool

File I/O Tools (Generic File I/O tool)

Level 0 Science Data Access Tools and Metadata tools

Coordinate System Conversion tool

Celestial Body Position (CBP) tool

Constant and Unit Conversion (CUC) tool

Geo-Coordinate Transformation (GCT) tool

Ancillary Data tool

Memory Management (MEM) tool

and Time and Date Conversion (TD) tools

Data - Sample science data files

Test Tools - Xrunner, Loadrunner, Compilers (FORTRAN77, FORTRAN90, and C), File Comparison tools, Toolkit Test drivers

4.6.1 Test Case 6.1: SCF Toolkit in SCF Environment Test (TS001.001)

This test case verifies that the SCF version of the PGS toolkit and the science software can be successfully compiled, linked and tested in the SCF environment and serves as a regression testing of the new/upgraded software. The SCF is the development environment for science software and the originator of all science related changes to a particular algorithm. Upon successful verification of the science software in the SCF environment, the new/upgraded software is available for integration into the DAAC. For this test, an integrated set of test drivers are used to represent the science software.

Test Inputs

Inputs to this test include calls to PGS toolkit functions. Parameters for these calls are contained in input files for each test driver.

Test Steps

See Section 5

Test Outputs

Outputs to this test include compilation status messages, logs, binary code, and toolkit return messages and data recorded in the test driver output files.

Success Criteria

Using the SCF version of the PGS toolkit, each science software delivery developed at the SCF should compile, link, and produce the expected test results in the SCF environment.

Assumptions and Constraints

None.

4.6.2 Test Case 6.2: SCF Toolkit in DAAC Environment Test (TS001.002)

This test case verifies that the SCF version of the PGS toolkit and the science software can be successfully compiled, linked and tested in the DAAC environment. Upon successful verification of the science software in the SCF environment, the new/upgraded software is made available for integration into the DAAC. The DAAC receiving the science software upgrade, compiles, links and tests the science software delivery with the SCF version of the PGS toolkit in the DAAC configuration. Test results are examined and compared with the expected test results delivered in the software delivery package. For this test, an integrated set of test drivers are used to represent the science software and the input and output files from Test Case 6.1 represents the test data delivered in the software delivery package.

Test Inputs

Inputs to this test include calls to PGS toolkit functions. Parameters for these calls are contained in input files for each test driver.

Test Steps

See Section 5

Test Outputs

Outputs to this test include compilation status messages, logs, binary code, and toolkit return messages and data recorded in the test driver output files.

Success Criteria

Using the SCF version of the PGS toolkit, each science software delivery developed at the SCF should compile, link, and produce the expected test results in the DAAC environment. Outputs from Test Case 6.1 are compared with Test Case 6.2 outputs to ensure the SCF version of the PGS toolkit provides consistent results in the SCF and DAAC environment.

Assumptions and Constraints

None.

4.6.3 Test Case 6.3: Test DAAC Toolkit in DAAC Environment (TS001.003)

This test case verifies that the DAAC version of the PGS toolkit and the science software can be successfully compiled, linked and tested in the DAAC environment. Upon successful verification of the science software in the DAAC environment using the SCF version of the PGS toolkit, the new/upgraded software is recompiled, linked and tested at the DAAC again, using the DAAC version of the PGS toolkit. Test results are examined and compared with the expected test results delivered in the software delivery package. For this test, an integrated set of test drivers are used to represent the science software and the input and output files from Test Case 6.1 represents the test data delivered in the software delivery package.

Test Inputs

Inputs to this test include calls to PGS toolkit functions. Parameters for these calls are contained in input files for each test driver.

Test Steps

See Section 5

Test Outputs

Outputs to this test include: compilation status messages and logs, binary code, and toolkit return messages and data recorded in the test driver output files.

Success Criteria

Using the DAAC version of the PGS toolkit, each science software delivery developed at the SCF should compile, link, and produce the expected test results in the DAAC environment. Outputs from Test Case 6.1 are compared with Test Case 6.3 outputs to ensure the DAAC version of the PGS toolkit provides results consistent with the SCF version.

Assumptions and Constraints

None.

4.6.4 Test Case 6.4: Successful Compile using SCF Version of the PGS Toolkit (B03.04.01)

This test verifies that the received algorithms in C, FORTRAN77 and Ada can be compiled and linked using the SCF version of the PGS toolkit. This test also verifies that memory leaks can be found using a dynamic analyzer. This is the first test in integrating an algorithm into the ECS system.

Test Inputs

SCF version of the PGS toolkit, science algorithms in C, FORTRAN77 and Ada.

Test Steps

See Section 5

Test Outputs

Binary executables, debugger report is displayed on the screen, number of bytes of memory leaked is reported if there are any, status messages and history log.

Success Criteria

The algorithms should be successfully compiled, linked, and the debugger reports no errors. The dynamic analyzer reports number of bytes of memory leaked if any. Appropriate status messages should be generated to reflect this activity and the history log updated.

Assumptions and Constraints

None.

4.6.5 Test Case 6.5: Unsuccessful Compile using SCF Version of the PGS Toolkit (B03.04.02)

This test verifies that the system responds to error conditions detected during algorithm compilation.

Test Inputs

SCF version of the PGS toolkit, science algorithms in C, FORTRAN77 and Ada containing compile errors.

Test Steps

See Section 5

Test Outputs

Debugger reports, status messages, history log.

Success Criteria

The system should report errors encountered during algorithm compilations and the history log should be updated to reflect these errors.

Assumptions and Constraints

None.

4.6.6 Test Case 6.6: Successful Compile using DAAC Version of the PGS Toolkit (B03.05.01)

This test verifies that received algorithms in C, FORTRAN77 and Ada can be debugged, compiled and linked on the DAAC platform using the DAAC version of the PGS toolkit. This test also verifies that memory leaks can be found using a dynamic analyzer. ANSI certified compilers/linkers are used for this test.

Test Inputs

DAAC version of the PGS toolkit, algorithms in C, FORTRAN77 and Ada to be ported to the DAAC platform.

Test Steps

See Section 5

Test Outputs

Binary executables, debugger report is displayed on the screen, number of bytes of memory leaked is reported if there are any, status messages, history log.

Success Criteria

The system should report on the successful linking, debugging and compiling on the DAAC platform. The dynamic analyzer reports number of bytes of memory leaked if any. The dynamic analyzer reports number of bytes of memory leaked if any. The history log should be updated.

Assumptions and Constraints

None.

4.6.7 Test Case 6.7: Unsuccessful Compile using DAAC Version of the PGS Toolkit (B03.05.02)

This test verifies errors encountered during compilation are reported. ANSI certified compilers/linkers are used for this test.

Test Inputs

DAAC version of the PGS toolkit, algorithms in C, FORTRAN77 and Ada containing compile errors.

Test Steps

See Section 5

Test Outputs

Debugger reports, status messages, history log.

Success Criteria

The system should report all compile errors encountered during compilation, and the history log should be updated to reflect such errors.

Assumptions and Constraints

None.

4.6.8 Test Case 6.8: ANSI Certified Compilers/Linkers Test (T03-01.07.01)

This test runs the makefiles and executes scripts for the compilation, loading and execution of science software provided by SCF. This is done for software developed in C, Ada and FORTRAN programming languages. Testing includes running algorithms which were successfully run at the SCF. Algorithms are tested on all approved DAAC platforms.

Test Inputs

Science software developed at the SCFs. Access to DAAC compilers and linkers. At least one algorithm developed in C, FORTRAN will be run. Algorithms without errors are used for input.

Test Steps

See Section 5

Test Outputs

Compiled code and status report.

Success Criteria

Code compiled and linked at the DAACs is compared to code compiled and linked at the SCF. No differences should be found between SCF compiled code and DAAC compiled code.

Assumptions and Constraints

None.

4.6.9 Test Case 6.9: ANSI Certified Compilers/Linkers Error Test (T03-01.07.02)

This test runs the makefiles and executes scripts for the compilation, loading and execution of science software provided by SCF. Testing includes running algorithms which contain known errors. This is done for software developed in C and FORTRAN programming languages. Algorithms are tested on all approved DAAC platforms.

Test Inputs

Science software developed at the SCFs. Access to DAAC compilers and linkers. At least one algorithm developed in C, FORTRAN will be run. Algorithms containing known errors are used for input.

Test Steps

See Section 5

Test Outputs

Status and error reporting.

Success Criteria

Algorithms containing errors should display complete and appropriate error messages.

Assumptions and Constraints

None.

4.6.10 Test Case 6.10: Standards Enforcement Test (T03-01.07.05)

In this test SCF science software, including algorithms and calibration coefficients are run against ECS standards checking software. EOSDIS standards checkers, which enforce ECS compliance, are run against the SCF developed science software. Standards checkers include code checkers, and any other tools which confirm compliance to ECS standards. SCF software is tested for completeness and correct format.

Test Inputs

Science software developed at the SCFs. Access to standards tools. Algorithms without errors are used for input.

Test Steps

See Section 5

Test Outputs

Status messages and error reporting

Success Criteria

Status is displayed. No error messages are displayed.

Assumptions and Constraints

None.

4.6.11 Test Case 6.11: Standards Enforcement Error Test (T03-01.07.06)

In this test SCF science software which is in violation of ECS standards, including algorithms and calibration coefficients, are run against ECS standards checking software. Standards checkers include code checkers, and any other tools which confirm compliance to ECS standards.

Test Inputs

Science software developed at the SCFs modified to include violations to ECS standards. Access to standards tools. Algorithms containing known errors are used for input.

Test Steps

See Section 5

Test Outputs

Status messages and error reporting

Success Criteria

Notification is displayed on the terminal screen of status and errors. All errors should be detected and status should state the error with enough detail to understand the nature of the non-compliance to ECS standards.

Assumptions and Constraints

None.

4.7 PDPS Thread Test

This thread demonstrates the capabilities provided by the Ir1 PDPS Processing Prototype. Namely two areas are implemented in Ir1: portions of the planning subsystem (for processing environment and resources) and the processing subsystem (data processing requests and COTS interfaces). Successful demonstration of the following objectives:

- Run single PGEs

- Support multiple PGEs in sequence
- Provide manual processing initiation and control
- Provide Release A processing interfaces
- Provide interfaces to CSS and MSS

Special resources required for this thread test include:

Hardware – Processing Queue Control workstations, representative DAAC Processing hosts

Software – Queuing and Execution Tool (Processing Management, Processing Queue Management), POSIX standards checker

Data – simulated CERES and LIS L0 data granule, simulated Product Generation Executable (PGE), Distributed Queuing System (DQS) and CHAINS interface are installed onto the machine which has MOTIF graphics libraries. One or more representative DAAC Processing hosts are set up as processing resources with DQS Exec installed.

Test Tools - Xrunner, Loadrunner

4.7.1 Test Case 7.1: Science Processing Documentation Test (TS003.008)

This test case demonstrates that Processing environment has either on-line display of documentation or printed documents for each tool installed. These documentation shall be displayed on console or in hard copy.

Test Inputs

Calls to invoke all on-line documentation for utilities and tools in processing environment

Test Steps

See Section 5

Test Outputs

On-line display or hard copy of documentation of each installed tool.

Success Criteria

For each tool installed, the existence of associated on-line and/or printed documentation will be displayed and verified by inspection.

Assumptions and Constraints

None.

4.7.2 Test Case 7.2: Science Processing Operating System and Utilities Test (TS003.006)

This test case verifies that the operating system for each UNIX platform in the science processing environment is in compliance with POSIX.2 standards and has C shell, Korn shell, and Bourne shell installed by running C shell, Korn shell and Bourne shell through the POSIX checker and invoking each shell. Utilities such as: perl, emacs, gzip, tar, imake, prof, gprof, nm, man, vi, make, lex and yacc are also available.

Test Inputs

UNIX commands to invoke C shell, Korn shell, Bourne shell, perl, emacs, gzip, tar, imake, prof, gprof, nm, man, vi, make, lex and yacc.

Test Steps

See Section 5

Test Outputs

Active POSIX.2 compliant C shell, Korn shell, and Bourne shell; status logs and messages

Success Criteria

For each representation of a Processing hardware/software environment, evidence of POSIX.2 compliance shall be shown and a functioning UNIX shells and utilities will be proven through logs and messages generated by the POSIX checker and status messages generated by invoking each command.

Assumptions and Constraints

None.

4.7.3 Test Case 7.3: Processing System Initialization and Shutdown Test (TS003.005)

This test demonstrates the ability to initialize and shutdown the Processing system in an orderly fashion. Using a system account with appropriate privileges, commands are entered to initialize the Processing system. The system status is monitored to determine the status of the initialization. Once the system is successfully initialized, commands are entered for orderly shutdown. Again, the system is monitored to determine shutdown status.

Test Inputs

A series of commands are entered for initialization, shutdown and system monitoring.

Test Steps

See Section 5

Test Outputs

System monitoring to confirm the state of the Processing system.

Success Criteria

The system commands are accepted and successful initialization and shutdown of the system is confirmed by system monitoring.

Assumptions and Constraints

None.

4.7.4 Test Case 7.4: Status of Data Processing Request (TS003.004)

This test verifies the capability to provide status of data processing request by continuously monitor the queue and processing resources. The basic information are: unstarted, queue, executing, complete, and error . They are displayed on the operator GUI.

Test Inputs

PGE chain is displayed on operator machine. One data processing request is allowed to successfully complete. Then cancellation requests are applied onto other data processing requests that are in queued and executing state.

Test Steps

See Section 5

Test Outputs

Operator GUI display of PGE chain shows the processing status. The status is dynamically updated.

Success Criteria

GUI display shows status of the data processing requests. The data processing requests are removed as the result of the cancellation requests. The one that is allowed to complete should have its status updated as complete.

Assumptions and Constraints

None.

4.7.5 Test Case 7.5: Unsuccessful Scheduling (B03.06.02)

This test verifies that the system responds to attempts to schedule algorithms which cannot be scheduled due to data dependencies.

Test Inputs

PDPS, algorithm which cannot be scheduled due to data dependencies.

Test Steps

See Section 5

Test Outputs

Status message, history log.

Success Criteria

The system should provide status messages reporting that the algorithm cannot be scheduled due to data dependencies. The history log should note that the attempt was unsuccessful.

Assumptions and Constraints

None.

4.7.6 Test Case 7.6: Suspending Execution of a Scheduled Processing Request (B03.06.03)

This test verifies that a scheduled processing request can be suspended. The tester will schedule a processing request to be executed. While it is executing, the tester will use the scheduling software to elect to suspend the executing processing request. A status message should be generated to indicate the processing request is suspended. Once it is suspended, the tester will verify its status.

Test Inputs

PDPS, display to observe processing request status, tool to suspend the executing processing request.

Test Steps

See Section 5

Test Outputs

Status message, display update.

Success Criteria

The display should update the status of the executing processing request to suspended, status message indicating the processing request is suspended.

Assumptions and Constraints

None.

4.7.7 Test Case 7.7: Resume Execution of a Scheduled Processing Request (B03.06.04)

This test will continue from test case 3 above. After the processing request is suspended and its status verified on a display, the tester will then use the scheduling software to resume the processing request. A status message should be generated to indicate the processing request is resumed. Once it is resumed, the tester will verify its status.

Test Inputs

Display to observe processing request status, tool to resume the suspended processing request.

Test Steps

See Section 5

Test Outputs

Status message, display update.

Success Criteria

The display should update the status of the suspended processing request to executing, status message indicating the processing request is executing.

Assumptions and Constraints

None.

4.7.8 Test Case 7.8: Canceling Execution of a Scheduled Processing Request (B03.06.05)

This test verifies that a scheduled processing request can be canceled. The tester will schedule a processing request to be executed. While it is executing, the tester will use the scheduling software to elect to cancel the executing processing request. A status message should be generated to indicate the processing request is canceled. Once it is canceled, the tester will verify its status.

Test Inputs

Display to observe processing request status, tool to cancel the executing processing request.

Test Steps

See Section 5

Test Outputs

Status message, display update.

Success Criteria

The display should update the status of the executing processing request to canceled, status message indicating the processing request is canceled.

Assumptions and Constraints

None.

4.7.9 Test Case 7.9: Queue (B03.06.06)

This test verifies the capability of changing the queue position of processing requests. The tester will manually change the position of a processing request in a queue after it has been scheduled for processing.

Test Inputs

Manually enter processing requests to activate an appropriate production plan to generate a data product. Change position of the requests in the queue for process.

Test Steps

See Section 5

Test Outputs

The tester can view the on-line status of requests in the processing queue. There will also be status in the message log.

Success Criteria

Changing the queue for processing requests will be logged in the processing log.

Assumptions and Constraints

None.

4.7.10 Test Case 7.10: Priority (B03.06.07)

This test verifies the capability to change request priorities. The tester will change the priority of a scheduled processing request. The system should verify this input and log the changed priority.

Test Inputs

Manually enter requests to activate an appropriate production plan to generate data product. Change the priorities of the requests in the processing queue.

Test Steps

See Section 5

Test Outputs

The tester can view the on-line status of requests in the processing queue. There will also be status in the message log.

Success Criteria

Changing priorities of requests in the processing queue will be logged in the processing log.

Assumptions and Constraints

None.

4.7.11 Test Case 7.11: Invalid Cancellation (B03.06.08)

This test verifies users authentication to cancel processing requests other than those generated by themselves. Several processing request will be scheduled by various testers. Then, the tester will try to cancel a request other than his own. The system should not allow this action and log the attempted transaction.

Test Inputs

Manually enter requests to activate appropriate production plans. Attempt to cancel the request of another tester after it begins execution.

Test Steps

See Section 5

Test Outputs

The tester can view the on-line status of requests in the processing queue. There will also be a status of the request in the message log.

Success Criteria

The request will not be terminated and an appropriate message will be logged.

Assumptions and Constraints

None.

4.7.12 Test Case 7.12: Invalid Priority Change (B03.06.09)

This test verifies that error messages are recorded in the processing log as a result of invalid change in request priority. Certain priorities will be reserved for identified requests. In this test, the tester will attempt to change the priority of his scheduled request beyond its allowed range. The system should respond to this request and log the attempt.

Test Inputs

Manually enter a request to activate an appropriate production plan. Modify the priority of the request after it begins execution.

Test Steps

See Section 5

Test Outputs

The tester can view the on-line status of requests in the processing queue. There will also be a status of the request in the message log.

Success Criteria

The priority will not be changed and an appropriate message will be logged.

Assumptions and Constraints

None.

4.7.13 Test Case 7.13: Invalid Change of Queue (B03.06.10)

This test verifies that error messages are recorded in the processing log as a result of invalid change of queue position. The tester will attempt to move his scheduled request in the queue to a position outside of the allowed range. The system should respond to this attempt.

Test Inputs

Manually enter a few requests to activate an appropriate production plan. Change the queue position of the requests that is currently being executed.

Test Steps

See Section 5

Test Outputs

The tester can view the on-line status of requests in the processing queue. There will also be a status for each request in the message log.

Success Criteria

The queue position will not be changed and an appropriate message will be logged.

Assumptions and Constraints

None.

4.7.14 Test Case 7.14: Data Processing Fault (B03.07.01)

This test verifies that the system management framework tool detects and locates the data processing machine fault and properly logs all events.

Test Inputs

Algorithm jobs. System management framework utility. Turn power off to the data processing machine.

Test Steps

TBD

Test Outputs

System management framework alerts tester of hardware fault. Algorithm processing will be halted.

Success Criteria

Event log records all activities of job processing. Event log should also list the job that were processing at the moment the data processing machine went down. After the system comes up again, the job status will change to "jobfailure".

Assumptions and Constraints

None.

4.7.15 Test Case 7.15: Queuing Fault (B03.07.02)

This test case has been merged with test case 4.7.14 (B03.07.01) to comply with Ir1's PDPS scope and objective.

4.7.16 Test Case 7.16: Planning Fault (B03.07.03)

This test case has been merged with test case 4.7.14 (B03.07.01) to comply with Ir1's PDPS scope and objective.

4.7.17 Test Case 7.17: PGS Resource Utilization Report Verification (B03.06.12)

After the above test cases in this thread have been successfully executed, the PGS resource utilization report will be printed and verified. The report should show the resources used to execute the processing requests for specific jobs. The ***autorep*** command will be used to list a variety of information about jobs and machines, and also list a summary of all currently defined jobs in the AutoSys Database.

Test Inputs

AutoSys jobs

SunOS platform

Test Steps

1) Initialize AutoSys on the platform provided.

2) Create a job through "Job Definition"

3) "Start Job"

4) Select Job

5) From the command window, issue the *autorep* command.

6) Run multiple *autorep* commands tha tare specified in the AutoSys Manual.

Test Outputs

PGS resource utilization report (*autorep*) print out.

Success Criteria

The PGS resource utilization report (*autorep*) should reflect some activities which occurred during this test thread.

Assumptions and Constraints

See AutoSys Manual page 15-13.

4.7.18 Test Case 7.18: PGS Processing Log Verification (B03.06.11)

This test will verify that the PGS processing log accounts for all the scheduling activities in this test thread. The tester will print the processing log after completing the above test cases in this thread. The processing log should contain correct information for some activities which occurred in this thread test. The *autolog* command wiil be used to view either the Event Processor log file or the specific Remote Agent log file for specified jobs.

Test Inputs

AutoSys

SunOS platform

Test Steps

1) Initialize AutoSys on the platform provided.

2) Create a job through "Job Definition"

3) "Start Job"

4) Select Job

5) From the command window, issue the *autolog* command.

6) Run multiple *autolog* commands that are specified in the AutoSys Manual.

Test Outputs

PGS processing log print out.

Success Criteria

The PGS processing log should reflect some activities which occurred during this thread test.

Assumptions and Constraints

See AutoSys Manual page 15-9.

4.8 Science Processing Suite Build Test

The Science Processing Suite Build Test represents an aggregation of the AI&T Tools, DAAC Toolkit, and PDPS threads. The functions to be tested include CM, logging, file transfer, reporting, and other AI&T and PDPS capabilities.

Special resources required for this build test include:

Hardware – Representative DAAC host (workstation or terminal), data storage device, printer, SCF toolkit hardware configuration, DAAC toolkit hardware configuration

Software – FTP, TCP/IP, e-mail, UNIX utilities, C, FORTRAN77, FORTRAN90, PGS toolkit tools, compilers, linkers, file comparison tools, report generation tools (word processors, spreadsheets, drawing tools), configuration management (ClearCase).

Data – science software (algorithm) delivery package, simulated Level 0 and ancillary data

Test Tools – Xrunner (optional), Loadrunner (optional), Static analyzer, memory leak detectors, custom ECS software (standards checkers, file comparison tool), scripts.

4.8.1 Test Case 8.1: Ingestion of Delivery Package (BS001.001)

This test case verifies that a Science Software delivery (Algorithm) package can be retrieved electronically from the Internet. Notification of delivery is accomplished by the receipt of an e-mail notification containing the File Transfer Protocol (FTP) address of the delivery package in the transfer directory. For this test, representations of delivery notifications, in a transfer directory, containing FTP addresses of Science Software delivery package locations are specified. The science software delivery packages are retrieved electronically via the Internet. A science software delivery package is retrieved in the form of a compressed tar file and should contain a delivery memo, documentation, software delivery, scripts, test plans, test data and expected test results. Upon successful receipt, the science software delivery is placed in the specified delivery directory and is untarred. Successful receipt of the delivery package is recorded, e-mail notification of receipt status is relayed and status log files are updated. The delivery package is validated, integrated, and tested in the simulated DAAC environment.

Test Inputs

Inputs to this test are valid e-mail Science software delivery notifications from the SCF to the DAAC.

Test Steps

See Section 5.

Test Outputs

Outputs to this test include fully transmitted delivery packages in the form of an uncompressed tar file and e-mail acknowledgments of successful or unsuccessful receipt.

Success Criteria

For valid science software delivery notification from the SCF, a delivery package is retrieved (compressed tar format), stored in the delivery directory at the DAAC and is uncompressed (if necessary) and untarred. Confirmation of receipt is successfully sent to the SCF.

Assumptions and Constraints

None.

4.8.2 Test Case 8.2: Evaluation, Inspection and Verification of the Science Software Delivery (BS001.002)

This test case verifies that a Science Software delivery package placed in non-public directory can be inspected for completeness and correct format (as specified in the delivery memo), and that the science software is compatible in the ECS environment. Verification of compatibility in the ECS environment encompasses proving that integration of the algorithm with currently running operational DAAC software interfaces can be performed, that preliminary performance statistics can be gathered, and that the mechanics to check for algorithm reliability and operational safety can be performed satisfactorily. Functionality that is confirmed in this test is the ability to display and print documentation provided in the delivery, to verify that all code/scripts deliveries are compliant with POSIX, ANSI and ESDIS standards and guidelines, that compilation capabilities exist, and that status logging can be performed throughout the entire process. For this test, representative science software deliveries in uncompressed file format are accepted, resulting in a configured and operationally acceptable science software delivery and an associated updated status logs.

Test Inputs

Inputs to this test include partial and complete science software deliveries in compressed tar format, deliveries that contain compliant and noncompliant software/scripts according to POSIX, ANSI and ESDIS standards.

Test Steps

See Section 5.

Test Outputs

Outputs to this test include configured, uncompressed, untarred science software delivery packages, supporting documentation in hard copy and on-line formats, and status logs.

Success Criteria

For each science software delivery package, all components listed in the delivery memo are accounted for, all software is POSIX, ANSI and ESDIS compliant (per the standards checkers), is successfully configured and status logs are produced of each transition.

Assumptions and Constraints

None.

4.8.3 Test Case 8.3: Science Software Delivery Integration Test (BS001.003)

This test case verifies that the portability of the science software (algorithm) delivery from the SCF environment to its perspective DAAC environment is achieved satisfactorily, that the system-level interfaces and value added features supplied by the PGS toolkit are functional in the DAAC environment through swapping of the SCF toolkit for the DAAC toolkit. The science software, delivery is compiled and linked with the SCF version of the PGS toolkit and the test plans provided in the delivery are executed. Results from this test are compared to the expected results supplied in the delivery. Upon successful completion of this phase, the science software delivery is compiled and linked with the DAAC version of the PGS toolkit. Results from this test are also compared to the expected results supplied in the delivery. Throughout this process, preliminary performance statistics and resource utilizations are gathered and status logs are produced. For this test representative science software deliveries will be used to verify the swapping of the SCF toolkit for the DAAC toolkit.

Test Inputs

Inputs to this test include compilable and non-compilable science software algorithms in the SCF toolkit environment and the DAAC toolkit environment.

Test Steps

See Section 5.

Test Outputs

Output to this test include a completely verified environment for the operational support of the science software at the DAAC, preliminary performance statistics, resource utilizations, and status logs.

Success Criteria

For each science software delivery, swapping of the SCF toolkit for the DAAC toolkit is done successfully, test plans supplied with the delivery are executed and expected results are received, preliminary performance statistics and resource utilizations are gathered and status logs are produced.

Assumptions and Constraints

None.

4.8.4 Test Case 8.4: Operational Testing of the Science Software Delivery (BS001.004)

This test case verifies that the science software runs safely within the Processing Subsystem, the product outputs produced in the SCF environment can also be produced in the DAAC environment and that the production configuration is correct. It should be noted here that although identical input values are used in the SCF and the DAAC, differences in hardware configuration or compilers may produce slightly different output values. This test verifies that operational transfer of the algorithm can be done successfully and that system related tests can be performed. For this test, representative science software will be run on multiple platforms, based on test plans provided with the science software delivery. The upgraded software will be run operationally as an Engineering version and labeled as "Unverified". Product output from the engineering version will be compared to the product output from the existing production version. Status reports are updated, performance, resource utilization information summary reports of the acceptance testing are logged to be used for evaluating the promotion of the science software to the operational phase.

Test Inputs

Inputs to this test are science software algorithms that are POSIX, ANSI and ESDIS compliant and have passed the success criteria for evaluation, inspection and verification.

Test Steps

See Section 5.

Test Outputs

Outputs to this test include a verified algorithm promoted for production processing, status messages, algorithm output products, acceptance testing summary report, and status logs.

Success Criteria

For each software algorithm submitted, the same shall be executed in the DAAC configuration and promoted to the operational phase.

Assumptions and Constraints

None.

4.8.5 Test Case 8.5: Algorithm Configuration Management (B03.02.01)

This test verifies that the received Science Software Delivery Package can be placed under configuration management. The configuration management tool must be able to place each element of the transfer package into its system.

Test Inputs

Configuration Management tool, the received algorithm package, request to place the received Science Software Delivery Package under configuration management.

Test Steps

See Section 5.

Test Outputs

Status message, history log updates.

Success Criteria

The algorithm should be placed under configuration management, appropriate status message should appear upon completion, the history log should reflect that the package is now under configuration management.

Assumptions and Constraints

None.

4.8.6 Test Case 8.6: Algorithm in Compliance with ECS Standards (B03.03.01)

This test involves running science software algorithms developed at a SCF against a POSIX checker to determine the software's ability to operate in the DAAC environment, and to ensure the source code is within ECS guidelines. An algorithm verified as having no known errors is run against the checker. This is done for algorithms in written in C and FORTRAN on all DAAC platforms.

Test Inputs

Science software developed at the SCFs. Access to the POSIX compliance checker. At least one algorithm developed in C and FORTRAN will be run on platforms representing all DAAC environments. Algorithms without errors are used for input.

Test Steps

See Section 5.

Test Outputs

Status messages, history log.

Success Criteria

Valid algorithms will return a status of compliance.

Assumptions and Constraints

None.

4.8.7 Test Case 8.7: Algorithm Not in Compliance with ECS Standards (B03.03.02)

This test involves running science software algorithms developed at a SCF against a POSIX checker to determine the software's ability to operate in the DAAC environment. An algorithm with known errors is tested (by rerunning the verified algorithm after changing POSIX API calls to non POSIX API calls) to confirm the checker's ability to detect POSIX non-compliance. This is done for algorithms in written in C and FORTRAN on all DAAC platforms.

Test Inputs

Science software developed at the SCFs. Access to the POSIX compliance checker. At least one algorithm developed in C and FORTRAN will be run on platforms representing all DAAC environments. Algorithms without errors and algorithms containing known errors are used for input.

Test Steps

See Section 5.

Test Outputs

Status messages and error reporting to history log.

Success Criteria

Invalid algorithms will return status of non-compliance. If non-compliance is found, all errors inserted in the representative algorithms should be detected.

Assumptions and Constraints

None.

4.8.8 Test Case 8.8: Algorithm Readiness Inventory (B03.03.03)

This test validates that the received algorithm characteristics are present prior to scheduling algorithm test time. This process is a manual process. The tester will record on the Inventory Log if the required characteristics are present. Any discrepancies will be noted in the Delivery Package Evaluation Report.

Test Inputs

Science Algorithm Delivery Package, Inventory Log.

Test Steps

See Section 5.

Test Outputs

Delivery Package Evaluation Report.

Success Criteria

The Inventory Log should include entries for the following characteristics of a received algorithm package: language used, operational impacts (e.g. algorithm software size, required resources), algorithm documentation, data handling standards, units used, models used, operational compatibility, and required metadata outputs. Any additional characteristics can be added to the log.

Assumptions and Constraints

None.

4.8.9 Test Case 8.9: Successful Scheduling and Processing of Algorithm (B03.06.01)

This test verifies the acceptance of the processing requests entered manually by the tester for generating Level 1A TRMM data products.

Test Inputs

Production plan containing science algorithm inputs, control parameters, resource validation, resource utilization, predicted processing times and simulated TRMM data products will be setup in the database. A Processing requests will be entered manually to activate the production plan.

Test Steps

See Section 5.

Test Outputs

Processing script, processing queues, status message, history log, results of algorithm execution.

Success Criteria

The PDPS should be able to schedule the science algorithms contained in the processing scripts to execute at the DAAC. These processing scripts are entered as jobs on the processing queues. The PDPS should be able to report on the schedule time, the success of execution, and the results of execution. The history log should be updated to reflect successful scheduling and execution.

Assumptions and Constraints

None.

4.8.10 Test Case 8.10: Verification of Event Log (B03.07.04)

This test verifies that the event log file records all system activities and algorithm jobs. Manually, the tester will review the event log file to verify that previous tests were completely recorded. The event log should contain correct information for the activities which occurred in this thread test.

Test Inputs

Previous test case activity.

Test Steps

See Section 5.

Test Outputs

Event log file print out.

Success Criteria

The event log file should reflect all activities which occurred during this thread test.

Assumptions and Constraints

None.

4.8.11 Test Case 8.11: Successful Comparison (B03.08.01)

This test verifies that the expected results provided in the Science Software Delivery Package match those generated during the actual execution of the algorithm.

Test Inputs

Comparison tool, expected results, actual results.

Test Steps

See Section 5.

Test Outputs

Status message, history log.

Success Criteria

The system should report on the successful comparison of the expected results with the actual results.

Assumptions and Constraints

None.

4.8.12 Test Case 8.12: Unsuccessful Comparison (B03.08.02)

This test verifies that the system reports differences when the expected results provided in the Science Software Delivery Package do not match those generated during the actual execution of the algorithm.

Test Inputs

Comparison tool, expected results, actual results (modified to provide errors).

Test Steps

See Section 5.

Test Outputs

Status message, history log.

Success Criteria

The system should report on the unsuccessful comparison of the expected results with the actual results.

Assumptions and Constraints

None.

4.8.13 Test Case 8.13: Successful Transfer (B03.09.01)

This test verifies that the system is able to send the algorithm test results to the SCF for analysis. The results should include: algorithm identification, test times, processor identification and test results.

Test Inputs

Actual test results for transferring from DAAC to SCF.

Test Steps

See Section 5.

Test Outputs

Fully transferred test result, status message and history log.

Success Criteria

The system should report on the successful transfer of the test results to the SCF.

Assumptions and Constraints

None.

4.8.14 Test Case 8.14: Unsuccessful Transfer (B03.09.02)

This test verifies that the system is able to respond to errors while attempting to send the algorithm test results to the SCF for analysis. An error will be induced to make the system fail during the transfer.

Test Inputs

Actual test results.

Test Steps

See Section 5.

Test Outputs

Status message, history log.

Success Criteria

The system should report on the failed attempt to transfer the test results to the SCF.

Assumptions and Constraints

None.

4.8.15 Test Case 8.15: Partial Transfer (B03.09.03)

This test verifies that the system is able to detect errors in content while trying to send the algorithm test results to the SCF for analysis. The results should include: algorithm identification, test times, processor identification and test results. The system should report if any of these elements are omitted from the transfer.

Test Inputs

Actual test results (modified to not contain all required elements).

Test Steps

See Section 5.

Test Outputs

Status message, history log.

Success Criteria

The system should report on the failed attempt due to incomplete packaging of the test results.

Assumptions and Constraints

None.

4.8.16 Test Case 8.16: Calibration Coefficients and Algorithm Update Test (B03.12.03)

This test verifies the DAAC's ability to receive and update science software using new or modified (updated) algorithms/calibration coefficients. This test will evaluate the ability of the DAAC toolkit environment to modified existing science software when an algorithm/coefficient update is deemed necessary by the SCF. An algorithm proven to be a valid algorithm from a previous test (Geolocation/Geocoordination Conversion Test or Time/Date Conversion Test) is modified and run at the SCF. The results are recorded. The same algorithm is modified at the DAAC using updated procedures and data received from the SCF. The results of the SCF output are compared to the DAAC output.

Test Inputs

Algorithms validated in previous tests such as the Geolocation/Geocoordination Conversion Test or Time/Date Conversion Test. New or updated algorithms and calibration coefficients. Access to all subroutines and libraries containing data needed to perform data conversions. Access to a script editor to make changes to the algorithms.

Test Steps

See Section 5.

Test Outputs

Algorithm status, algorithm product output for both the SCF toolkit and the DAAC toolkit.

Success Criteria

The product outputs from the SCF and DAAC toolkits are analyzed for correct scientific product content. There should be similar output from the SCF and DAAC toolkits. Outputs from the SCF and DAAC toolkits may be examined using a proven data comparison tool.

Assumptions and Constraints

None.

4.8.17 Test Case 8.17: Software Problem Reports (B03.13.01)

This test verifies the ECS’s capability to send a Software Problem Report to the ECS Science Community. The Software Problem Report will contain the reason for updating and/or replacing the algorithms running at an appropriate DAAC.

Test Inputs

Write a Software Problem Report at an appropriate DAAC. This report should contain the reason for updating the algorithm running at that DAAC. Transmit this report to the ECS Science Community.

Test Steps

See Section 5.

Test Outputs

The ECS Science Community will receive the Software Problem Report from the DAAC.

Success Criteria

The content of the Software Problem Report received at the science community should be exactly the same as it was sent from the DAAC.

Assumptions and Constraints

None.

4.8.18 Test Case 8.18: Integration Support Request (B03.13.02)

This test verifies the ECS’s capability to send for receiving an integration support request from the ECS Science Community. An integration support request will be to replace and/or update the algorithms running at an appropriate DAAC.

Test Inputs

Management policies and procedures integration support guidelines from the EDF.

Test Steps

See Section 5.

Test Outputs

Management policies and procedures manual from GSFC, MSFC, and LaRC.

Success Criteria

Guidelines within document are current and identical to that copy held at the EDF.

Assumptions and Constraints

None.

4.8.19 Test Case 8.19: I/O to Intermediate Storage (B03.14.02)

An analysis will be performed for appropriate hardware and software to ensure that the PGS has the capacity to support I/O to intermediate storage as required by individual science algorithms.

Test Inputs

Support documentation for analysis.

Test Steps

See Section 5.

Test Outputs

A detailed analysis report.

Success Criteria

Presented in RMA Analysis Report.

Assumptions and Constraints

None.

4.8.20 Test Case 8.20: Multiple Passes over Input Product (B03.13.03)

An analysis will be performed for appropriate hardware and software to ensure that the PGS has the capacity to support multiple passes over input products as required by individual science algorithms.

Test Inputs

Support documentation for analysis.

Test Steps

See Section 5.

Test Outputs

A detailed analysis report.

Success Criteria

Presented in RMA Analysis Report.

Assumptions and Constraints

None.

4.8.21 Test Case 8.21: Build and Install CM Platform (T03-01.01.01)

This test verifies the ability to establish a workstation (platform) as the repository for the CM Tool and all the associated source code, libraries, and executable for algorithms, and COTS packages.

Test Inputs

CM Tool Software package, source code, libraries, and executables for science algorithms, and COTS packages. Checksum values for all delivered source code, libraries, executables, etc. Network environment to support file transfers or tape I/O capability.

Test Steps

See Section 5.

Test Outputs

An established CM repository, appropriate status/error messages and log entries generated as a result of building/installing the CM workstation.

Success Criteria

All checksums for source code, libraries, executables etc. should match those provided by CM.

Assumptions and Constraints

None.

4.8.22 Test Case 8.22: Building Executable Code for an ECS Defined Platform (T03-01.02.01)

This test verifies the ability to build executable code for an ECS defined platform.

Test Inputs

Installed CM repository workstation with source code, libraries, and make files to support creation of executable codes for the ECS defined platform.

Test Steps

See Section 5.

Test Outputs

Executable code for the ECS defined platform, appropriate status/error messages and log entries generated as a result of executing make files.

Success Criteria

Valid executable codes ready to be installed on the ECS defined platform. Executable checksum values for one ECS defined platform are not expected to match those of a different ECS defined platform type.

Assumptions and Constraints

None.

4.8.23 Test Case 8.23: Build/Install Previous Version of an Executable Code (T03-01.04.02)

This test demonstrates the ability to build and install a previous version (other than the current version number) of an executable code on a required platform.

Test Inputs

All of the previous versions of source code and libraries required to build an older version of an executable code, the configuration record used to build the previous binary and the platform type the binary is being built for. Network environment to support file transfers or Tape I/O capability to support installation of the executable code.

Test Steps

See Section 5.

Test Outputs

Appropriate status/error messages and log entries generated as a result of building and installing the previous version of the executable code.

Success Criteria

Tester was able to find and select the configuration record used to build the previous version of the executable code using the CM tool, build the executable code, install it on the required platform, and verify that it will execute on the platform after installation.

Assumptions and Constraints

None.

4.8.24 Test Case 8.24: Geolocation/Geocoordination Conversion Test (T03-01.09.01)

This test verifies that an algorithm developed and deemed valid at the SCF, will run correctly giving the same algorithm output as produced in the SCF when run at the DAAC. Representative SCF algorithms that utilizes PGS toolkit functions for geolocation/geocoordinate transformations are run on all approved DAAC platforms. The results are analyzed to determine the scientific correctness of the data products.

Test Inputs

Algorithms developed at the SCF with the SCF version of the PGS toolkits. Access to all subroutines and libraries containing data needed to perform data conversions. These algorithms will use the PGS toolkit for geolocation/geocoordinate transformations. The algorithm is first run at the DAAC using the PGS SCF toolkit version. Then the algorithm is run at the DAAC using the DAAC toolkit version.

Test Steps

See Section 5.

Test Outputs

Algorithm status, algorithm product output for both the SCF toolkit and the DAAC toolkit.

Success Criteria

The product outputs from the SCF and DAAC toolkits are analyzed for correct scientific product content. There should be similar output from the SCF and DAAC toolkits. Outputs from the SCF and DAAC toolkits may be examined using a proven data comparison tool.

Assumptions and Constraints

None.

4.8.25 Test Case 8.25: Time/Date Conversion Test (T03-01.09.02)

This test verifies that an algorithm developed and deemed valid at the SCF, will run correctly giving the same algorithm output as produced in the SCF when run at the DAAC. Representative SCF algorithms that utilizes PGS toolkit functions for time/date conversions are run on all approved DAAC platforms. The results are analyzed to determine the scientific correctness of the data products.

Test Inputs

Algorithms developed at the SCF with the SCF version of the PGS toolkits. Access to all subroutines and libraries containing data needed to perform data conversions. These algorithms will use the PGS toolkit for time/date conversions. The algorithm is first run at the DAAC using the PGS SCF toolkit version. Then the algorithm is run at the DAAC using the DAAC toolkit version.

Test Steps

See Section 5.

Test Outputs

Algorithm status, algorithm product output for both the SCF toolkit and the DAAC toolkit.

Success Criteria

The product outputs from the SCF and DAAC toolkits are analyzed for correct scientific product content. There should be similar output from the SCF and DAAC toolkits. Outputs from the SCF and DAAC toolkits may be examined using a proven data comparison tool.

Assumptions and Constraints

None.

4.8.26 Test Case 8.26: Calibration Coefficients and Algorithm Update Test (T03-01.09.03)

This test verifies the DAAC's ability to receive and update science software using new or modified (updated) algorithms/calibration coefficients. This test will evaluate the ability of the DAAC toolkit environment to modified existing science software when an algorithm/coefficient update is deemed necessary by the SCF. An algorithm proven to be a valid algorithm from a previous test (Geolocation/Geocoordination Conversion Test or Time/Date Conversion Test) is modified and run at the SCF. The results are recorded. The same algorithm is modified at the DAAC using updated procedures and data received from the SCF. The results of the SCF output are compared to the DAAC output.

Test Inputs

Algorithms validated in previous tests such as the Geolocation/Geocoordination Conversion Test or Time/Date Conversion Test. New or updated algorithms and calibration coefficients. Access to all subroutines and libraries containing data needed to perform data conversions. Access to a script editor to make changes to the algorithms.

Test Steps

See Section 5.

Test Outputs

Algorithm status, algorithm product output for both the SCF toolkit and the DAAC toolkit.

Success Criteria

The product outputs from the SCF and DAAC toolkits are analyzed for correct scientific product content. There should be similar output from the SCF and DAAC toolkits. Outputs from the SCF and DAAC toolkits may be examined using a proven data comparison tool.

Assumptions and Constraints

None.

4.8.27 Test Case 8.27: Remote AI&T Access for SCFs (BS001.005)

The objective of the Remote AI&T Access for SCFs test is to demonstrate the ability of remote users to log into the DAAC environment and access the various AI&T tools, services, and applications required for algorithm integration and test of the SCF software.

Test Inputs

Inputs to this test case include 3-5 remote usernames and passwords (with assigned privileges for all AI&T tools short of making changes to tools developed for Ir1), and sample algorithms for evaluation.

Test Steps

Remotely log into DAAC from SCF workstation using multiple accounts

Utilize AI&T tools, services, and applications as part of the SCF algorithm integration and test within the DAAC facility (both in interactive and batch modes)

Terminate remote session with DAAC AI&T environment

Test Outputs

The expected outputs include the respective tool log outputs and compiled versions of the SCF algorithms for each username/account.

Success Criteria

This test will be deemed successful when the SCF algorithms can be remotely integrated and tested within the DAAC facility.

Assumptions and Constraints

Remote access can also be expected while local AI&T is occurring and/or during off-hours (while ECS M&O personnel are not supporting the Ir1 DAAC). In addition, certain degradation in the remote performance might occur during X-window sessions.

Examples of tools accessed remotely by SCFs include:

- Code Checkers (FORTRAN 77, SPARCworks for Sun, CASEVision for SGI)
- Standards Checker
- Compilers (on Sun and SGI platforms)
- File Comparison Tools
- Event Log Queries for Reports (Sybase)

Examples of Ir1 components not accessed remotely by SCFs:

- OpenView
- ClearCase
- Production Prototype

- OA Tools
- AI&T Manager
- Toolkit Libraries

4.9 TRMM SDPF Ingest Thread Test

This thread verifies the capability of the ECS at the LaRC and MSFC DAACs to receive and verify Authentication Requests, DANs, and data files from the SDPF.

Special resources required for this thread test include:

- o TRMM I/F Simulator
- o XRunner
- o Data Comparison Tools

4.9.1 Test Case 9.1: SDPF Authentication Request with Valid ID Test (TS004.001)

This test demonstrates the ability of the ECS at the LaRC and MSFC DAACs to receive a valid Authentication Request from the SDPF and correctly verify if the request is sent from an approved authorized source. Upon successfully establishing a communications connection, the SDPF sends an Authentication Request to the ECS. The request is a formatted message sent using UNIX sockets. The request includes identification (ID) data. The ECS receives the Authentication Request and determines that the originator of the request is authorized by checking the ID data. A response is sent using UNIX sockets to the SDPF indicating the Authentication Request as valid and therefore the established connection is accepted.

<u>Test Inputs</u>

A series of Authentication Requests in correct format (as described in the ICD between the SDPF and TRMM consumers) with valid ID data.

<u>Test Steps</u>

See Section 5

<u>Test Outputs</u>

Authentication Responses indicating acceptance of the Authentication Requests are sent to the SDPF and are entered into the event log.

<u>Success Criteria</u>

A communications connection is successfully established and all Authentication Requests sent by the SDPF are received and correctly determined to have valid ID data. Each Authentication Request is answered by a response indicating the acceptance of the request and therefore acceptance of the connection and are also entered into the event log.

Assumptions and Constraints

Special configuration issues:

Hardware – ingest workstation, client host (simulated SDPF)

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications, CSMS Event Logging

Simulator for the SDPF interface and the X-Runner Tool to record the test

4.9.2 Test Case 9.2: SDPF Authentication Requests with Invalid ID Test (TS004.003)

This test demonstrates the ability of the ECS at the LaRC and MSFC DAACs to receive and recognize an invalid Authentication Request from the SDPF and correctly verify that the request is sent from an unauthorized source. Upon successfully establishing a communications connection, the SDPF sends an Authentication Request to the ECS. The request is a formatted message sent using UNIX sockets. The request includes invalid ID data. The ECS receives the Authentication Request and determines that the originator of the request is not authorized by checking the ID data. A response is sent using UNIX sockets to the SDPF indicating the Authentication Request is invalid and therefore the established connection is rejected.

Test Inputs

A series of Authentication Requests in correct format (as described in the ICD between the SDPF and TRMM consumers) with invalid ID data (ID data not recognized by the DAAC as that of a valid user).

Test Steps

See Section 5

Test Outputs

Authentication Responses indicating rejection of the Authentication Requests are sent to the SDPF and are entered into the event log.

Success Criteria

A communications connection is successfully established and all Authentication Requests sent by the SDPF are received and correctly determined to have invalid ID data. Each Authentication Request is answered by a response indicating the rejection of the request and therefore rejection of the connection and are also entered into the event log.

Assumptions and Constraints

Special configuration issues:

Hardware – ingest workstation, client host (simulated SDPF)

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications, CSMS Event Logging

Simulator for the SDPF interface and the X-Runner Tool to record the test

4.9.3 Test Case 9.3: SDPF Valid Data Availability Notice Verification Test (TS004.005)

This test demonstrates the ability of the ECS at the LaRC and MSFC DAACs to receive Data Availability Notices from the SDPF and validate the notices for adherence to ECS standards. Data ingest is initiated when the data provider sends a Data Availability Notice (DAN) informing the ECS that data is available for ingest. A DAN contains a header, linked to data files. The header contains information about the linked files. The linked data files describe a data product using Consultative Committee for Space Data Systems (CCSDS) standards. One DAN may describe one data file, or several data files, that are available for ingest. The ECS does a validation check on the "EXPIRATION_TIME" keyword in the PVL portion of the DAN to verify that the date/time prior to which the data will remain available is a valid date/time (i.e. EXPIRATION_TIME is identified as a future time not a past time). After validation of the DAN a Data Availability Acknowledgment (DAA) message is sent to the SDPF.

Test Inputs

Inputs to this test include a series of DANs submitted electronically using a SDPF simulated interface to the ECS. This includes DANs with a single detached header linked to a single file and DANs with single detached headers linked to multiple files. Only DANs with valid "EXPIRATION_TIME" information are submitted.

Test Steps

See Section 5

Test Outputs

Outputs to this test are a DAA for each DAN received.

Success Criteria

This test is deemed successful if each DAN submitted is received and validated correctly, and a DAA is sent to the SDPF indicating that each DAN is valid.

Assumptions and Constraints

Special configuration issues:

Hardware – ingest workstation, client host (simulated SDPF)

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications

Simulator for the SDPF interface and the X-Runner Tool to record the test

4.9.4 Test Case 9.4: SDPF Invalid Data Availability Notice Verification Test (TS004.007)

This test demonstrates the ability of the ECS at the LaRC and MSFC DAACs to recognize Data Availability Notices from the SDPF, that do not adhere to ECS standards, as invalid. Data ingest is initiated when the data provider sends a DAN informing the ECS that data is available for ingest. A DAN contains a header, linked to data files. The header contains information about the linked files. The linked data files describe a data product using CCSDS standards. One DAN may describe one data file, or several data files, that are available for ingest. The ECS does a validation check on the "EXPIRATION_TIME" keyword in the PVL portion of the DAN to verify that the date/time prior to which the data will remain available is a valid date/time. For this test all DANs submitted contain invalid date/time tags indicating data availability. The ECS determines that the "EXPIRATION_TIME" information in the DAN is invalid and a message is sent to the SDPF indicating the disposition of the DAN as invalid. The message indicates the reason for declaring a DAN as invalid.

Test Inputs

Inputs to this test include a series of erroneous DANs are submitted electronically using a SDPF simulated interface to the ECS. Only DANs with invalid data/time tags indicating data availability are submitted.

Test Steps

See Section 5

Test Outputs

Outputs to this test are status messages sent from the ECS to the data provider (SDPF) for each DAN received.

Success Criteria

This test is deemed successful if each DAN submitted is received and validated correctly, and a status message is sent in response, indicating that the "EXPIRATION_TIME" information in the PVL portion of the DAN as invalid.

Assumptions and Constraints

Special configuration issues:

Hardware – ingest workstation, client host (simulated SDPF)

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications

Simulator for the SDPF interface and the X-Runner Tool to record the test.

4.9.5 Test Case 9.5: SDPF FTP-Get Single File Data Ingest Test (TS005.001)

This test demonstrates the ability of the ECS at the LaRC and MSFC DAACs to ingest a data collection containing a single data file of TRMM data. A DAN is sent from a simulated SDPF interface and is received by the ECS, indicating the availability of data for ingest. An interim capability for file transfer is available for early interface file transfer testing. The file indicated in the DAN is retrieved via FTP. The data is placed on temporary magnetic storage.

Test Inputs

Inputs to this test include: Two DANs, one DAN to contain a single file of CERES QL data and the other DAN to contain a single file of LIS QL data.

Test Steps

See Section 5

Test Outputs

Outputs to this test include: Authentication Responses, DAAs, DDNs, Event Logs, and temporary storage directory listings.

Success Criteria

This test is deemed successful if the Authentication Responses indicate a disposition of "accepted" and for each DAN submitted from the SDPF the DDN indicates a disposition of "successful". All DANs received, result in successful retrieval of data from the SDPF to the ECS. All appropriate entries are made in the Event Log.

Assumptions and Constraints

Special configuration issues:

Hardware – ingest workstation, client host (simulated SDPF), storage device

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for Ir1 only), CSMS Interprocess Communications, CSMS Event Logging

Data – simulated CERES and LIS QL data.

Simulator for the SDPF interface and the X-Runner Tool to record the test.

4.9.6 Test Case 9.6: SDPF FTP-Get and Multiple File Ingest Test (TS005.003)

This test case demonstrates the ability of the ECS at the LaRC and MSFC DAACs to ingest multiple TRMM data files. A DAN is received by the ECS. The DAN contains a header, linked to a product specification for multiple data files. The DAN contains data set identification, and data granule identification. An interim capability for file transfer is available for early interface file transfer testing. Data is transferred and placed in an ECS directory.

Test Inputs

Inputs to this test include: Two DANs, one DAN to contain at least one file of CERES L0 data and one file each of TRMM Predictive and Definitive Orbit data and the other DAN to contain at least one file of LIS L0 data and one file each of TRMM Predictive and Definitive Orbit data for multiple file ingest.

<u>Test Steps</u>

See Section 5

<u>Test Outputs</u>

Outputs to this test include: Authentication Responses, DAAs, DDNs, Event Logs, and temporary storage directory listings.

<u>Success Criteria</u>

This test is deemed successful if the Authentication Responses indicate a disposition of "accepted" and for each DAN submitted from the SDPF the DDN indicates a disposition of "successful". All DANs received, result in successful retrieval of data from the SDPF to the ECS. All appropriate entries are made in the Event Log.

<u>Assumptions and Constraints</u>

Special configuration issues:

Hardware – ingest workstation, client host (simulated SDPF), storage device

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for Ir1 only), CSMS Interprocess Communications, CSMS Event Logging

Data – simulated CERES and LIS L0 data and TRMM Predictive and Definitive Orbit data.

Simulator for the SDPF interface and the X-Runner Tool to record the test.

4.10 TRMM TSDIS Ingest Thread Test

This thread verifies the capability of the ECS at the GSFC and MSFC DAACs to receive and verify Authentication Requests, DANs, and data files from the TSDIS.

Special resources required for this thread test include:

- TRMM I/F Simulator
- XRunner
- Data Comparison Tools

4.10.1 Test Case 10.1: TSDIS Authentication Requests with Valid ID Test (TS004.002)

This test demonstrates the ability of the ECS at the GSFC and MSFC DAACs to receive a valid Authentication Request from the TSDIS and correctly verify if the request is sent from an approved authorized source. Upon successfully establishing a communications connection, the TSDIS sends an Authentication Request to the ECS. The request is a formatted message sent using UNIX sockets. The request includes ID data. The ECS receives the Authentication Request and determines that the originator of the request is authorized by checking the ID data. A response is sent using UNIX sockets to the TSDIS indicating the Authentication Request is valid and therefore the established connection is accepted.

Test Inputs

A series of Authentication Requests in correct format (as described in the ICD between the ECS and TSDIS) with valid ID data.

Test Steps

See Section 5

Test Outputs

Authentication Responses indicating acceptance of the Authentication Requests are sent to the TSDIS and are entered into the event log.

Success Criteria

A communications connection is successfully established and all Authentication Requests sent by the TSDIS are received and correctly determined to have valid ID data. Each Authentication Request is answered by a response indicating the acceptance of the request and are also entered into the event log.

Assumptions and Constraints

Special configuration issues:

Hardware – ingest workstation, client host (simulated TSDIS)

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications, CSMS Event Logging

Simulator for the TSDIS interface and the X-Runner Tool to record the test.

4.10.2 Test Case 10.2: TSDIS Authentication Request with Invalid ID Test (TS004.004)

This test demonstrates the ability of the ECS at the GSFC and MSFC DAACs to receive and recognize an invalid Authentication Request from the TSDIS and correctly verify that the request is sent from an unauthorized source. Upon successfully establishing a communications connection, the TSDIS sends an Authentication Request to the ECS. The request is a formatted message sent using UNIX sockets. The request includes invalid ID data. The ECS receives the Authentication Request and determines that the originator of the request is not authorized by checking the ID data. A response is sent using UNIX sockets to the TSDIS indicating the Authentication Request is invalid and therefore the established connection is rejected.

Test Inputs

A series of Authentication Requests in correct format (as described in the ICD between the ECS and TSDIS) with invalid ID data (ID data not recognized by the DAAC as that of a valid user).

Test Steps

See Section 5

Test Outputs

Authentication Responses indicating rejection of the Authentication Requests are sent to the TSDIS and are entered into the event log.

Success Criteria

A communications connection is successfully established and all Authentication Requests sent by the TSDIS are received and correctly determined to have invalid ID data. Each Authentication Request is answered by a response indicating the rejection of the request and therefore rejection of the connection and are also entered into the event log.

Assumptions and Constraints

Special configuration issues:

Hardware – ingest workstation, client host (simulated TSDIS)

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications, CSMS Event Logging

Simulator for the TSDIS interface and the X-Runner Tool to record the test.

4.10.3 Test Case 10.3: TSDIS Valid Data Availability Notice Verification Test (TS004.006)

This test demonstrates the ability of the ECS at the GSFC and MSFC DAACs to receive Data Availability Notices from the TSDIS and validate the notices for adherence to ECS standards. Data ingest is initiated when the data provider sends a DAN informing the ECS that data is available for ingest. A DAN contains a header, linked to data files. The header contains information about the linked files. The linked data files describe a data product using CCSDS standards. One DAN may describe one data file, or several data files, that are available for ingest. The ECS does a validation check on the "EXPIRATION_TIME" keyword in the PVL portion of the DAN to verify that the date/time prior to which the data will remain available is a valid date/time (i.e. EXPIRATION_TIME is identified as a future time not a past time). After validation of the DAN a DAA is sent to the TSDIS.

Test Inputs

Inputs to this test include a series of DANs submitted electronically using a TSDIS simulated interface to the ECS. This includes DANs with a single detached header linked to a single file and DANs with single detached headers linked to multiple files. Only DANs with valid "EXPIRATION_TIME" information are submitted.

Test Steps

See Section 5

Test Outputs

Outputs to this test are a DAA for each DAN received.

Success Criteria

This test is deemed successful if each DAN submitted is received and validated correctly, and a DAA is sent to the TSDIS indicating that each DAN is valid.

Assumptions and Constraints

Special configuration issues:

Hardware – ingest workstation, client host (simulated TSDIS)

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications

Simulator for the TSDIS interface and X-Runner Tool to record the test.

4.10.4 Test Case 10.4: TSDIS Invalid Data Availability Notice Verification Test (TS004.008)

This test demonstrates the ability of the ECS at the GSFC and MSFC DAACs to recognize Data Availability Notices from the TSDIS, that do not adhere to ECS standards, as invalid. Data ingest is initiated when the data provider sends a DAN informing the ECS that data is available for ingest. A DAN contains a header, linked to data files. The header contains information about the linked files. The linked data files describe a data product using CCSDS standards. One DAN may describe one data file, or several data files, that are available for ingest. The ECS does a validation check on the "EXPIRATION_TIME" keyword in the PVL portion of the DAN to verify that the date/time prior to which the data will remain available is a valid date/time . For this test all DANs submitted contain invalid date/time tags indicating data availability. The ECS determines that the "EXPIRATION_TIME" information in the DAN is invalid and a status message is sent to the TSDIS, indicating the disposition of the DAN as invalid. The message indicates the reason for declaring a DAN as invalid.

Test Inputs

Inputs to this test include a series of erroneous DANs submitted electronically using a TSDIS simulated interface to the ECS. Only DANs with invalid date/time tags indicating data availability are submitted.

Test Steps

See Section 5

Test Outputs

Outputs to this test are status messages sent from the ECS to the data provider (TSDIS) for each DAN received.

Success Criteria

This test is deemed successful if each DAN submitted is received and validated correctly, and a status message is sent in response, indicating that the "EXPIRATION_TIME" information in the PVL portion of the DAN as invalid.

Assumptions and Constraints

Special configuration issues:

Hardware – ingest workstation, client host (simulated TSDIS)

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, CSMS Interprocess Communications

Simulator for the TSDIS interface the X-Runner Tool to record the test

4.10.5 Test Case 10.5: TSDIS FTP-Get Single File Data Ingest Test (TS005.002)

This test demonstrates the ability of the ECS at the GSFC and MSFC DAACs to ingest a data collection containing a single data file of TRMM data. A DAN is sent from a simulated TSDIS interface and is received by the ECS, indicating the availability of data for ingest. An interim capability for file transfer is available for early interface file transfer testing. The file indicated in the DAN is retrieved via FTP. The data is placed on temporary magnetic storage.

Test Inputs

Inputs to this test include: Five DANs, each DAN to contain a single unique file of either PR, TMI, VIRS Level 1A data or GV Level 1B data or VIRS combined data .

Test Steps

See Section 5

Test Outputs

Outputs to this test include: Authentication Responses, DAAs, DDNs, Event Logs, and temporary storage directory listings.

Success Criteria

This test is deemed successful if the Authentication Responses indicate a disposition of "accepted" and for each DAN submitted from the TSDIS the DDN indicates a disposition of "successful". All DANs received, result in successful retrieval of data from the TSDIS to the ECS. All appropriate entries are made in the Event Log.

Assumptions and Constraints

Special configuration issues:

Hardware – ingest workstation, client host (simulated TSDIS), storage device

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for Ir1 only), CSMS Interprocess Communications, CSMS Event Logging

Data – simulated PR, TMI, VIRS Level 1A data, GV Level 1B data, VIRS combined data.

Simulator for the TSDIS interface and the X-Runner Tool to record the test.

4.10.6 Test Case 10.6: TSDIS FTP-Get and Multiple File Ingest Test (TS005.004)

This test case demonstrates the ability of the ECS at the GSFC and MSFC DAACs to ingest multiple TRMM data files. A DAN is received by the ECS. The DAN contains a header, linked to a product specification for multiple data files. The DAN contains data set identification, and data granule identification. An interim capability for file transfer is available for early interface file transfer testing. Data is transferred and placed in an ECS directory.

Test Inputs

Inputs to this test include: Two DANs, one DAN to contain VIRS L1A-1B data and browse data and the other to contain PR, TMI, GV Level 1A-3B data and browse data and VIRS combined data and browse data for multiple file ingest.

Test Steps

See Section 5

Test Outputs

Outputs to this test include: Authentication Responses, DAAs, DDNs, Event Logs, and temporary storage directory listings.

Success Criteria

This test is deemed successful if the Authentication Responses indicate a disposition of "accepted" and for each DAN submitted from the TSDIS the DDN indicates a disposition of "successful". All DANs received, result in successful retrieval of data from the TSDIS to the ECS. All appropriate entries are made in the Event Log.

Assumptions and Constraints

Special configuration issues:

Hardware – ingest workstation, client host (simulated TSDIS), storage device

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for Ir1 only), CSMS Interprocess Communications, CSMS Event Logging

Data – simulated PR, TMI, GV and VIRS data.

Simulator for the TSDIS interface and the X-Runner Tool to record the test.

4.11 Data Server Interface Thread Test

This thread verifies the capability of the ECS to receive and verify Authentication and Data Requests from TSDIS and from TSUs.

Special resources required for this thread test include:

- ECS Gateway I/F
- Data Server I/F
- Data Consumer Simulator
- XRunner
- Data Comparison Tools

4.11.1 Test Case 11.1: Authentication Requests with Valid ID Test (TS006.001)

This test demonstrates the ability of the ECS Gateway to receive and recognize a valid Authentication Request from a client and correctly verify if the request is sent from an approved authorized source. Upon successfully establishing a communications connection, the client sends an Authentication Request to the ECS Gateway . The request is a formatted message sent using UNIX sockets and includes ID data. The ECS Gateway acknowledges the receipt of the request and determines that the originator of the request is authorized by checking the ID data. A response is sent using UNIX sockets to the client indicating the request is valid and therefore the established connection is accepted.

Test Inputs

A series of Authentication Requests with valid IDs.

Test Steps

See Section 5

Test Outputs

Responses indicating acceptance of the Authentication Requests are logged.

Success Criteria

A communications connection is successfully established and all Authentication Requests sent by the client are received and correctly determined to have valid IDs. Each request is answered by a response indicating the acceptance of the request and therefore acceptance of the connection.

Assumptions and Constraints

None

4.11.2 Test Case 11.2: Authentication Request with Invalid ID Test (TS006.002)

This test demonstrates the ability of the ECS Gateway to receive and recognize an invalid Authentication Request from a client and correctly verify that the request is sent from an unauthorized source. Upon successfully establishing a communications connection, the client sends an Authentication Request to the ECS Gateway. The request is a formatted message sent using UNIX sockets and includes invalid ID data. The ECS Gateway acknowledges the receipt of the request and determines that the originator of the request is not authorized by checking the ID data. A response is sent using UNIX sockets to the client indicating the request is invalid and therefore the established connection is rejected.

Test Inputs

A series of Authentication Requests with IDs not recognized by the ECS Gateway as a valid user.

Test Steps

See Section 5

Test Outputs

Responses indicating rejection of the Authentication Requests are logged.

Success Criteria

A communications connection is successfully established and all Authentication Requests sent by the clients are received and correctly determined to have invalid IDs. Each Authentication Request is answered by a response indicating the rejection of the request and therefore rejection of the connection.

Assumptions and Constraints

None

4.11.3 Test Case 11.3: Data Request Test - Data Available(TS006.003)

This test demonstrates the ability of the ECS Gateway to receive and recognize a Data Request from a client, send the request to the Data Server to be executed, and return a Data Availability Notice (DAN) from the Data Server to the client. Upon successfully establishing a connection, the client sends a Data Request to the ECS Gateway. The request is a formatted message sent using TCP and includes Data_Type, Instrument_Name, etc.. A connection is made to the Data Server which executes all commands within the Data Request. The Data Server returns to the ECS Gateway a status indicating that the command has completed and a parameter list containing the available files for the requested data. The ECS Gateway sends the DAN to the client.

Test Inputs

A series of Data Requests with valid inputs.

Test Steps

See Section 5

Test Outputs

Responses indicating acceptance and execution of the Data Requests are logged.

Success Criteria

A connection is successfully established and all Data Requests sent by the client are received and executed by the Data Server. Each Data Request is answered by a DAN indicating a status and a pointer to the data.

Assumptions and Constraints

None

4.11.4 Test Case 11.4: Data Request Test - Data Not Available(TS006.004)

This test demonstrates the ability of the ECS Gateway to receive and recognize a Data Request from a client, send the request to the Data Server to be executed, and return a Data Availability Notice (DAN) from the Data Server to the client. Upon successfully establishing a connection, the client sends a Data Request to the ECS Gateway. The request is a formatted message sent using TCP and includes Data_Type, Instrument_Name, etc.. A connection is made to the Data Server which executes all commands within the Data Request. The Data Server returns to the ECS Gateway a status indicating that the command has completed and a parameter list containing no available files for the requested data. The ECS Gateway sends the DAN to the client.

Test Inputs

A series of Data Requests with invalid inputs.

Test Steps

See Section 5

Test Outputs

Responses indicating acceptance and execution of the Data Requests are logged.

Success Criteria

A connection is successfully established and all Data Requests sent by the client are received and executed by the Data Server. Each Data Request is answered by a DAN indicating no files available for the requested data.

Assumptions and Constraints

None

4.12 TRMM Interface Build Test

The TRMM Interface Build Test represents an aggregation of the TRMM SDPF Ingest, TRMM TSDIS Ingest, and Data Server threads. The functions to be tested also include the generation of polling requests.

Special resources required for this build test include:

- XRunner
- TRMM I/F Simulator
- Data Comparison Tools

4.12.1 Test Case 12.1: SDPF FTP-Get File Validation and Ingest Test (BS002.001)

This test demonstrates the ability to establish a network connection between the SDPF and the ECS at the LaRC and MSFC DAACs for ingest of a series of data collections, including collections containing single and multiple granules of data. The SDPF sends an Authentication Request to the ECS. The ECS verifies the Authentication Request as from a valid source and a connection is established allowing the SDPF to send DANs. The DANs are received and validated. DAAs are sent to the SDPF. Upon validation, the data is retrieved from the SDPF. Data Delivery Notices (DDNs) are sent to the data providers to indicate successful transfer of the data. Data is placed temporarily on disk storage media.

Test Inputs

Inputs to this test include: DANs, at least one data file for each data type to include CERES and LIS QL and L0 and TRMM Predictive and Definitive Orbit data for single and multiple file ingest.

Test Steps

See Section 5

Test Outputs

Outputs to this test include: Authentication Responses, DAAs, DDNs, Event Logs, temporary storage directory listings and data comparison results.

Success Criteria

A data connection is successfully established allowing the SDPF to send DANs to the ECS. DAAs for each DAN are sent to the SDPF. For each DAN, the data is successfully retrieved and placed on storage media. DDNs are sent to the SDPF. Data comparison of the data before ingest to data after ingest, shows no significant differences.

Assumptions and Constraints

Special configuration issues:

Hardware – ingest workstation, client host (simulated SDPF), storage device

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software only for Ir1), CSMS Interprocess Communications, CSMS Event Logging

Data – simulated CERES and LIS QL and L0 data and TRMM Predictive and Definitive Orbit data.

Simulator for the SDPF interface, data comparison tool, and the X-Runner Tool to record the test.

4.12.2 Test Case 12.2: TSDIS FTP-Get File Validation and Ingest Test (BS002.002)

This test demonstrates the ability to establish a network connection between the TSDIS and the ECS at the GSFC and MSFC DAACs for ingest of a series of data collections, including collections containing single and multiple granules of data. the TSDIS sends an Authentication Request to the ECS. The ECS verifies the Authentication Requests as from a valid source and a connection is established allowing the TSDIS to send DANs. The DANs are received and validated. DAAs are sent to the TSDIS. Upon validation, the data is retrieved from the TSDIS. DDNs are sent to the data providers to indicate successful transfer of the data. Data is placed temporarily on disk storage media.

Test Inputs

Inputs to this test include: DANs, at least one data file for each data type and level of data to include PR, TMI, GV and VIRS data for single and multiple file ingest.

Test Steps

See Section 5

Test Outputs

Outputs to this test include: Authentication Responses, DAAs, DDNs, Event Logs, temporary storage directory listings and data comparison results.

Success Criteria

A data connection is successfully established allowing the SDPF to send DANs to the ECS. DAAs for each DAN are sent to the TSDIS. For each DAN, the data is successfully retrieved and placed on storage media. DDNs are sent to the TSDIS. Data comparison of the data before ingest to data after ingest, shows no significant differences.

Assumptions and Constraints

Special configuration issues:

Hardware – ingest workstation, client host (simulated TSDIS), storage device

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for Ir1 only), CSMS Interprocess Communications, CSMS Event Logging

Data – simulated PR, TMI, GV and VIRS data (all Levels).

Simulator for the TSDIS interface, data comparison tool, and the X-Runner Tool to record the test.

4.12.3 Test Case 12.3: SDPF Status Reporting Test (BS002.003)

This test demonstrates the ability to send status messages to SDPF and to enter error conditions into the Event Log. Required status messages to be generated by ECS and sent to SDPF Status messages are as follows:

- file transfer failure
- file size discrepancies
- invalid data type identifier
- missing required metadata
- metadata parameters out of range
- data conversion failure
- failure to archive data
- inability to transfer data in a specified time window
- missing required request information
- successful archive of data

Error conditions that are required to be logged are as follows:

- receipt of an unexpected message
- detection of invalid information in a message (validation of DAN_SEQ_NO only)
- communications failure with provider of Ingest Request
- file transfer failures
- Discrepancies between number of files received and the specifications in the Ingest Request

Test Inputs

Inputs to this test include ingest requests from the SDPF. At least one ingest request is submitted to test for each instance of status reporting and error logging.

Test Steps

See Section 5

Test Outputs

Outputs to this test include status messages sent to the SDPF and event logs.

Success Criteria

This test is deemed successful if all ingest requests submitted are received and for each ingest request received an appropriate status message is sent to SDPF or the appropriate entry is made to the event log.

Assumptions and Constraints

Special configuration issues:

Hardware – workstation, terminal, ingest client host (simulated SDPF)

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for Ir1 only), CSMS Interprocess Communications, CSMS Event Logging

Data – simulated TRMM data.

Simulator for SDPF interface and the X-Runner Tool to record the test.

4.12.4 Test Case 12.4: TSDIS Status Reporting Test (BS002.004)

This test demonstrates the ability to send status messages to TSDIS and to enter error conditions into the Event Log. Required status messages to be generated by ECS and sent to TSDIS Status messages are as follows:

- file transfer failure
- file size discrepancies
- invalid data type identifier
- missing required metadata
- metadata parameters out of range
- data conversion failure
- failure to archive data
- inability to transfer data in a specified time window

- missing required request information
- successful archive of data

Error conditions that are required to be logged are as follows:

- receipt of an unexpected message
- detection of invalid information in a message (validation of DAN_SEQ_NO only)
- communications failure with provider of Ingest Request
- file transfer failures
- Discrepancies between number of files received and the specifications in the Ingest Request

Test Inputs

Inputs to this test include ingest requests from the TSDIS. At least one ingest request is submitted to test for each instance of status reporting and error logging.

Test Steps

See Section 5

Test Outputs

Outputs to this test include status messages sent to the TSDIS and event logs.

Success Criteria

This test is deemed successful if all ingest requests submitted are received and for each ingest request received an appropriate status message is sent to TSDIS or the appropriate entry is made to the event log.

Assumptions and Constraints

Special configuration issues:

Hardware – workstation, terminal, ingest client host (simulated TSDIS)

Software – Ingest Client Interface CSC, Automated Network Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for Ir1 only), CSMS Interprocess Communications, CSMS Event Logging

Data – simulated TRMM data.

Simulator for TSDIS interface and the X-Runner Tool to record the test.

4.12.5 Test Case 12.5: NESDIS and GDAO Polling Ingest Test (BS002.005)

This test case demonstrates the ability to initiate ingest by polling a location for the presence of TRMM ancillary data files and upon detection of a data granule generate a Polling Request. This Polling Request indicates the location of the data files. The data files identified in the Polling

Request are transferred and the data is placed in ECS storage. Data comparison of the data before ingest to data after ingest, shows no significant differences.

Test Inputs

Inputs to this test include detection and transfer of TRMM ancillary data files.

Test Steps

See Section 5

Test Outputs

Outputs to this test include entries into a log indicating generation of a Polling Request, temporary storage directory listings and data comparison results.

Success Criteria

This test is deemed successful if files are detected as a result of polling, and an Ingest Polling Request is generated. Each Polling Request is assigned a unique id and contains the location of the data. Data is successfully transferred. Data comparison of the data before ingest to data after ingest shows no significant differences.

Assumptions and Constraints

Special configuration issues:

Hardware – workstation, terminal, ingest client host (simulated NESDIS and GDAO), storage devices

Software - Ingest Client Interface CSC, Polling Ingest Client Interface CSC, Ingest Working File Collection CSC (test software for Ir1 only), CSMS Interprocess Communications, CSMS Event Logging

Data – simulated NESDIS and GDAO ancillary data.

Ancillary data at NESDIS and GDAO in a location accessible to the ESN, data comparison tool and the X-Runner Tool to record the test.

4.13 Ir1 Release Build Test

The Ir1 Release Build Test represents an aggregation of the System Administration, Science Processing Suite, and TRMM Interface Build tests. In addition to being conducted at the end of the SI&T cycle, these test cases will also comprise the core group of tests which will be used for regression testing and remote site testing at the DAACs.

Special resources required for this build test include:

- Cell Directory Service Command Program (cdscp)
- XRunner
- LoadRunner

- Privileges to execute DTS management functions
- Sample science data files
- HP OpenView
- Network analyzer
- TRMM I/F Simulator

Table 4.13-1. Site Regression Test Suite (1 of 2)

Functional Area	Section ID	Test Case ID
Inter-Site Communications and Interfaces		
Ingest	4.12.1	BS002.001
	4.12.2	BS002.002
	4.12.5	BS002.005
Data Server	4.11.3	TS006.003
	4.11.4	TS006.004
Infrastructure	4.1.17	T01-02.02.01
CSS (ftp, e-mail, BB,rcp)	4.2.2	BC012.004
	4.2.7	TC010.002
	4.2.9	TC010.003
	4.2.15	T01-02.04.01
	4.2.16	T01-02.04.02
	4.2.18	T01-02.04.04
	4.2.19	T01-02.04.05
	4.2.20	T01-02.04.06
	4.2.22	T01-02.04.08
	4.2.23	TC009.003
	4.2.26	B01.07.01
System Monitoring		
HPOV	4.3.13	TC014.002
	4.3.26	T04-01.05.01
Inspection		
system Inspection	4.1.1	TC017.001
Security	4.1.11	B01.01.02
	4.1.17	T01-02.02.01
Fault Isolation	4.13.11	B03.11.01
	4.13.12	B03.11.02
	4.13.13	B03.11.03
	4.13.14	B03.11.04
	4.2.28	T01-02.05.07
Reports & Event Logs	4.3.19	BC016.003
	4.3.9	TC013.003
	4.3.11	TC013.005
CM	4.13.1	T03-01.01.01
	4.13.2	T03-01.02.01
	4.13.3	T03-01.04.01

Table 4.13-1. Site Regression Test Suite (2 of 2)

Functional Area	Section ID	Test Case ID
System Adminstration		
Access	4.4.5	T01-02.02.02
	4.4.6	T01-02.02.03
	4.4.8	BC002.002
AI &T Tools		
Code standard Checking	4.5.5	TS002.006
Load Testing		
Processing & Scheduling	4.13.4	B03.10.01
	4.13.5	B03.10.02
	4.13.6	B03.10.03
	4.13.7	B03.10.04
	4.13.8	B03.10.05
	4.13.9	B03.10.06
	4.13.10	B03.10.07
	4.8.3	BS001.003
	4.8.4	BS001.004
Remote AI&T Access	4.8.27	BS001.005
	4.8.10	B03.07.04
Ingest		variations of simultaneous tests
Data Server		variations of simultaneous tests
Toolkits		variations of simultaneous tests

4.13.1 Test Case 13.1: Build and Install CM Platform (T03-01.01.01)

This test verifies the ability to establish a workstation (platform) as the repository for the CM Tool and all the associated source code, libraries, and executable for algorithms, and COTS packages.

Test Inputs

CM Tool Software package, source code, libraries, and executables for science algorithms, and COTS packages. Checksum values for all delivered source code, libraries, executables, etc. Network environment to support file transfers or tape I/O capability.

Test Steps

See Section 5

Test Outputs

An established CM repository, appropriate status/error messages and log entries generated as a result of building/installing the CM workstation.

Success Criteria

All checksums for source code, libraries, executables etc. should match those provided by CM.

Assumptions and Constraints

None.

4.13.2 Test Case 13.2: Building, Installing, and Verifying Executable Code for an ECS Defined Platform (T03-01.02.01)

This test verifies the ability to build, install, and verify executable code for an ECS defined platform.

Test Inputs

Installed CM repository workstation with source code, libraries, and make files to support creation of executable codes for the ECS defined platform. Installation procedures/scripts required to support the platform install. COTS packages that must be installed. Network environment to support file transfers or Tape I/O capability.

Test Steps

See Section 5

Test Outputs

Executable code for the ECS defined platform, appropriate status/error messages and log entries generated as a result of executing make files and software installation.

Success Criteria

Valid executable codes ready to be installed on the ECS defined platform. Executable checksum values for one ECS defined platform are not expected to match those of a different ECS defined platform type. An ECS defined platform capable of performing the functions/operations provided as part of the Ir1 system release.

Assumptions and Constraints

None.

4.13.3 Test Case 13.3: Build/Install Previous Version of an Executable Code (T03-01.04.02)

This test demonstrates the ability to build and install a previous version (other than the current version number) of an executable code on a required platform.

Test Inputs

All of the previous versions of source code and libraries required to build an older version of an executable code, the configuration record used to build the previous binary and the platform type the binary is being built for. Network environment to support file transfers or Tape I/O capability to support installation of the executable code.

Test Steps

See Section 5

Test Outputs

Appropriate status/error messages and log entries generated as a result of building and installing the previous version of the executable code.

Success Criteria

Tester was able to find and select the configuration record used to build the previous version of the executable code using the CM tool, build the executable code, install it on the required platform, and verify that it will execute on the platform after installation.

Assumptions and Constraints

None.

4.13.4 Test Case 13.4: Ancillary Data Access - Digital Terrain Map DB (B03.10.01)

This test verifies the capability to access and use a digital terrain map database.

Test Inputs

PGS Toolkit and AA data with suitable test driver and test driver input data file will be used to perform this test.

Test Steps

See Section 5

Test Outputs

The tester can view the output result produced by the test driver and compare the output result with the sample result file.

Success Criteria

The comparison of the result should report the differences in machine name, outfile name, time and date.

Assumptions and Constraints

None.

4.13.5 Test Case 13.5: Ancillary Data Access - Land/Sea DB (B03.10.02)

This test verifies the capability to access and use a land/sea database .

Test Inputs

PGS Toolkit and AA data with suitable test driver and test driver input data file will be used to perform this test.

Test Steps

See Section 5

Test Outputs

The tester can view the output result produced by the test driver and compare the output result with the sample result file.

Success Criteria

The comparison of the result should report the differences in machine name, outfile name, time and date.

Success Criteria

An acceptance of request will be logged.

Assumptions and Constraints

None.

4.13.6 Test Case 13.6: Ancillary Data Access - Climatology DB (B03.10.03)

This test verifies the capability to access and use a climatology database.

Test Inputs

PGS Toolkit and AA data with suitable test driver and test driver input data file will be used to perform this test.

Test Steps

See Section 5

Test Outputs

The tester can view the output result produced by the test driver and compare the output result with the sample result file.

Success Criteria

The comparison of the result should report the differences in machine name, outfile name, time and date.

he tester can view the on-line status of requests in the processing queue. There will also be a status of request in the message log.

Success Criteria

An acceptance of request will be logged.

Assumptions and Constraints

None.

4.13.7 Test Case 13.7: Ancillary Data Access - Digital Political Map DB (B03.10.04)

This test verifies the capability to access and use a digital political map database.

Test Inputs

PGS Toolkit and AA data with suitable test driver and test driver input data file will be used to perform this test.

Test Steps

See Section 5

Test Outputs

The tester can view the output result produced by the test driver and compare the output result with the sample result file.

Success Criteria

The comparison of the result should report the differences in machine name, outfile name, time and date.

The tester can view the on-line status of requests in the processing queue. There will also be status of requests in the message log.

Success Criteria

An acceptance of request will be logged.

Assumptions and Constraints

None.

4.13.8 Test Case 13.8: TRMM Ancillary Data Access (B03.10.05)

This test verifies the capability to access TRMM ancillary data products by polling a location for the presence of TRMM ancillary data files and upon detection of a data granule generate a polling request. The data files identified in the polling request are transferred and the data is place in ECS storage. This test was found to be redundant with part of testcase 12.5(NESDIS and GADO Polling Ingest Test - BS002.005). "Comparison of the data before ingest and data after ingest" is not part of this test case.

Test Inputs

Inputs to this test include location and detection of TRMM ancillary data files.

Test Steps

See Section 5

Test Outputs

The tester can view the result of polling requests in the log.

Success Criteria

This test is deemed successful if files are detected as a result of polling and data successfully transferred.

Assumptions and Constraints

None.

4.13.9 Test Case 13.9: Missing Data (B03.10.06)

This test verifies that the processing will not success until data is available. Before the driver will run the tester will remove the necessary ancillary data file. When the execution begins, the production should be suspended due to data availability. The tester will then replace the data which was removed. At this time, the tester will resume execution of the product generation.

Test Inputs

PGS Toolkit and AA data with suitable test driver and test driver input data file will be used to perform this test.

Test Steps

See Section 5

Test Outputs

The tester can view the output result produced by the test driver and compare the output result with the sample result file.

Success Criteria

The request will not be processed until the data become available. These activities will be logged.

Assumptions and Constraints

None.

4.13.10 Test Case 13.10: Erroneous Data (B03.10.07)

This test verifies that the processing will not success if the input data is incorrect. Incorrect data file will need to be inputted into the driver.

Test Inputs

PGS Toolkit and AA data with suitable test driver and test driver input data file will be used to perform this test.

Test Steps

See Section 5

Test Outputs

The tester can view the on-line status of the process.

Success Criteria

The request will not be processed once the erroneous data is discovered. This activity will be logged.

Assumptions and Constraints

None.

4.13.11 Test Case 13.11: Local Area Network Fault (B03.11.01)

This test verifies that the system management framework tool will properly detect and locate the network fault that occurs during the transfer of Science Software Delivery Packages. The network fault will occur within the LAN of a representative DAAC host in the EDF. This test verifies that proper alarms are displayed and the fault is recorded in event logs.

Test Inputs

SCF notification of intent to deliver a Science Software Delivery Package. Root map of system management framework tool displayed on tester's machine. Turn power off (reset) on gateway/router into DAAC LAN.

Test Steps

See Section 5

Test Outputs

File transfer is terminated. Symbols on root/sub map of System Management Framework has changed.

Success Criteria

Software delivery terminated. Traversing through the system management framework maps the tester should be directed to the faulty gateway/router. Log file should record all system and tester activities.

Assumptions and Constraints

None.

4.13.12 Test Case 13.12: Wide Area Network Fault (B03.11.02)

This test verifies that the system management framework tool will properly detect and locate the network fault that occurs during the transfer of Science Software Delivery Packages. The tester will simulate the network fault within the connection of the WAN to the LAN.

Test Inputs

SCF notification of intent to deliver a Science Software Delivery Package. Root map of system management framework tool displayed on tester's machine. Connection of WAN to LAN disconnected.

Test Steps

See Section 5

Test Outputs

File transfer terminated. Symbol on root/sub map of system management framework has changed.

Success Criteria

Software delivery terminated. Traversing through the system management framework maps the tester should be directed to the problem. Log file should record all system and tester activities.

Assumptions and Constraints

None.

4.13.13 Test Case 13.13: Host Machine Fault (B03.11.03)

This test verifies that the system management framework tool will properly detect and locate the host machine within the EDF that caused the simulated network fault. The host machine within the EDF where the software is being delivered will be shutdown, during this process the transfer should be terminated and the system management framework should notify tester of fault.

Test Inputs

SCF notification of intent to deliver a Science Software Delivery Package. Root map of system management framework tool displayed on testers machine. Shut down host machine where software delivery is targeted.

Test Steps

See Section 5

Test Outputs

File transfer terminated. DAAC/host symbol of root/sub map of system management framework has changed.

Success Criteria

Software delivery terminated. Traversing through the system management framework maps the tester should be directed to the fault host machine. Log file should record all system and tester activities.

Assumptions and Constraints

None.

4.13.14 Test Case 13.14: File Transfer Termination (B03.11.04)

This test verifies that the system management framework tool will properly detect a file transfer termination. Testing includes monitoring the system management framework tool while the file transfer process is located and terminated.

Test Inputs

SCF notification of intent to deliver a Science Software Delivery Package. Root map of system management framework tool displayed on tester's machine. Locate and terminate file transfer process from SCF.

Test Steps

See Section 5

Test Outputs

File transfer terminated abruptly. Symbol of root/sub map of system management framework has changed.

Success Criteria

Software delivery terminated. Traversing through the system management framework maps the tester should be directed to the transfer fault. Log file should record all system and tester activities.

Assumptions and Constraints

None.